

IBM DB2 通用数据库



Net Search Extender 管理和用户指南

版本 8.1

IBM DB2 通用数据库



Net Search Extender 管理和用户指南

版本 8.1

注意！

在使用本资料及其支持的产品之前，请务必阅读第 285 页的附录 O，『声明』中的一般信息。

第三版，2003 年 6 月

本版本适用于 DB2 通用数据库 Net Search Extender 版本 8.1（程序号为 5765-F38）及所有后续发行版和修订版，直到在新版本中另有声明为止。

本版本将替换 SH12-6740-01。

© Copyright International Business Machines Corporation 1995, 2003. All rights reserved.

目录

关于本书	vii
本书读者	vii
如何使用本书	vii
如何阅读语法图	viii
相关信息	viii
如何发送意见	ix
与 IBM 联系	ix
产品信息	x
更改摘要	xiii

第 1 部分 用户指南 1

第 1 章 概述和概念	3
关键概念	3
使用 SQL 标量搜索函数	6
使用存储过程搜索	7
使用 SQL 表值函数来进行搜索	8
附加概念	8
列变换函数	9
实例服务	9
外部存储的数据	9
管理视图	9
分区数据库支持	9
对联合数据库中的昵称建立索引	10
关键功能	10
DB2 客户机 / 服务器环境中的 DB2 Net Search Extender	11
第 2 章 安装	13
系统需求	13
分区 DB2 服务器的安装概述 (UNIX)	14
在 UNIX 上安装	14
适用于 UNIX 的步骤 1: 安装产品组件	14
适用于 UNIX 的步骤 2: 更新 DB2 实例	15
Windows 安装	15
目录名和文件名	15
安装 Outside-In 库	16
安装验证	16
Windows 上的安装验证	16
UNIX 上的安装验证	16

第 3 章 用户方案	19
使用 SQL 标量搜索函数的简单示例	19
使用高速缓存用法和存储过程搜索的简单示例	21
使用 SQL 表值函数的简单示例	22
第 4 章 规划	25
目录位置和索引存储器	25
文档格式和受支持的代码页	25
Outside-In 过滤软件	26
User 角色	27
使用命令行或“DB2 控制中心”界面来建立索引	28

第 5 章 Net Search Extender 实例服务	29
启动和停止 DB2 Net Search Extender	29
锁定服务	29
使用锁定服务	30
查看锁定快照	32
更新服务	32

第 6 章 创建和维护文本索引	35
介绍 db2text 命令	35
启用数据库	37
禁用数据库	38
创建文本索引	39
对二进制数据类型创建文本索引	41
对不受支持的数据类型创建文本索引	41
对 DATALINK 数据类型创建文本索引	42
安装 Data Links jar 文件	43
使用增量索引更新 (它使用“DB2 复制”) 来对昵称创建文本索引	43
创建存储过程搜索可以使用的文本索引	44
有关视图的文本索引	48
性能注意事项	50
维护文本索引	50
更新和重组文本索引	51
改变文本索引	52
清除索引事件	53
删除文本索引	54
查看文本索引状态	54
备份与复原索引	55

第 7 章 使用 DB2 控制中心	57	CONTROL	108
启动和停止“DB2 Net Search Extender 实例服务”	58	START	110
启用数据库	59	STOP	111
文本索引管理	60	第 12 章 数据库管理员的管理命令	113
创建文本索引	61	ENABLE DATABASE	114
维护文本索引	73	DISABLE DATABASE	116
第 8 章 搜索	83	DB2EXTDL (实用程序)	118
使用 SQL 标量搜索函数搜索文本	84	DB2EXTHL (实用程序)	119
进行查询	84	第 13 章 文本表所有者的管理命令	121
搜索并返回找到的匹配项数目	84	ACTIVATE CACHE	122
搜索并返回找到的文本文档的分数	85	ALTER INDEX	124
指定 SQL 搜索自变量	85	CLEAR EVENTS	128
按任意顺序搜索术语	86	CREATE INDEX	130
使用布尔运算符 AND 和 OR 进行搜索	86	DEACTIVATE CACHE	145
模糊搜索	87	DROP INDEX	147
搜索术语的部分 (字符屏蔽)	87	DB2EXTTH (实用程序)	149
搜索已经包含屏蔽字符的术语	88	UPDATE INDEX	151
按固定顺序搜索术语	88	HELP	155
在同一句子或段落中搜索术语	88	COPYRIGHT	157
在结构化文档的各节中搜索术语	88	第 14 章 搜索自变量的语法	159
使用布尔运算符 NOT 进行搜索	89	搜索自变量	160
同义字搜索	89	第 15 章 SQL 标量搜索函数和 SQL 表值函数	167
数字属性搜索	89	搜索函数总结	167
自由文本搜索	90	CONTAINS	168
附加搜索语法示例	90	NUMBEROFMATCHES	169
使用存储过程搜索来搜索文本	91	SCORE	170
使用“SQL 表值函数”来搜索文本	92	DB2EXT.TEXTSEARCH	171
使用 HIGHLIGHT 函数	92	DB2EXT.HIGHLIGHT	175
搜索多列	94	第 16 章 存储过程搜索函数	179
性能注意事项	95	DB2EXT.TEXTSEARCH (对于存储过程搜索)	180
第 9 章 使用结构化文档	97	具有 SQL 查询的存储过程	183
第 10 章 使用同义字来扩展搜索术语	99	输入参数	183
同义字的结构	99	输入 / 输出参数	183
预定义同义字关系	100	输出参数	183
定义您自己的关系	101	输出结果集	184
创建和编译同义字	101	第 17 章 结构化文档支持	185
创建同义字定义文件	101	文档模型	185
将定义文件编译为同义字典	103	缺省文档模型	185
第 2 部分 参考	105	为结构化纯文本文档定义文档模型	186
第 11 章 实例所有者的管理命令	107		

元素参数	187
对 GPP 文档建立索引时的情况	188
为 HTML 文档定义文档模型	189
元素参数	190
为 XML 文档定义文档模型	191
元素参数	192
对 XML 文档建立索引时的情况	194
为 Outside-In 过滤的文档定义文档模型	195
元素参数	196
对 Outside-In 文档建立索引时的情况	196

第 18 章 同义字支持	197
-------------------------------	------------

第 3 部分 附录	201
----------------------------	------------

附录 A. 迁移	203
---------------------------	------------

从 Net Search Extender V8.1 移至 Net Search Extender V8.1.x.	203
从 Net Search Extender V7.2 移至 Net Search Extender V8.1.x.	204
从 Text Information Extender V7.2 移至 Net Search Extender V8.1.x	204

附录 B. 使用大量内存	207
-------------------------------	------------

AIX (32 位和 64 位)	207
Windows (32 位)	207
Solaris 操作环境 (32 位和 64 位)	208
Linux (32 位)	209
HP-UX (32 位和 64 位)	209

附录 C. Net Search Extender 信息目录	211
---	------------

数据库级别信息的视图	211
db2ext.proxyinformation 表	213
索引级别信息的视图	213
db2ext.textindexes 视图	214
db2ext.indexconfiguration 视图	215
db2ext.textindexformats 视图	216
文本索引的表视图	217
事件视图	217
日志表、视图和昵称	218

附录 D. 受支持的 CCSID	221
-----------------------------------	------------

CCSID	221
-----------------	-----

附录 E. 受支持的语言	229
-------------------------------	------------

附录 F. Net Search Extender 消息	233
---	------------

信息和警告消息	233
错误消息	233

附录 G. 文档模型参考	253
-------------------------------	------------

文档模型的 DTD	253
定位器 (XPath) 表达式的语义	254
文本字段和文档属性的限制	256
Outside-In 标记属性值	257

附录 H. 文本搜索引擎	261
-------------------------------	------------

标记化	261
词语	261
句子	261
段落	261
无用词	261
语言支持无用词	262

附录 I. 文本搜索引擎原因代码	263
-----------------------------------	------------

附录 J. 故障诊断	271
-----------------------------	------------

跟踪故障	271
----------------	-----

附录 K. Data Link 消息	273
-------------------------------------	------------

附录 L. 同义字支持的 CCSID	275
-------------------------------------	------------

CCSID	275
-----------------	-----

附录 M. 同义字工具返回的消息	277
-----------------------------------	------------

附录 N. Windows 系统错误	281
-------------------------------------	------------

系统错误	281
----------------	-----

附录 O. 声明	285
---------------------------	------------

商标	287
--------------	-----

词汇表	289
----------------------	------------

索引	293
---------------------	------------

关于本书

本书描述如何使用 IBM DB2 Universal Database™ Net Search Extender 来准备和维护 DB2® 数据库以检索文本数据。它还描述可以如何使用所提供的 SQL 函数来存取和处理这些类型的数据。通过将 DB2 Net Search Extender 的函数组合到程序的 SQL 语句中，可以创建功能强大的通用文本检索程序。

在本书中提到的“DB2”就是指 DB2 UDB。

本书读者

本书供熟悉 DB2 管理概念、工具和技术的 DB2 数据库管理员使用。

本书还供熟悉 SQL 和可用于 DB2 应用程序的一种或多种编程语言的 DB2 应用程序员使用。

如何使用本书

本书的结构如下：

『第一部分 用户指南』

此部分概述了 DB2 Net Search Extender，描述如何安装和设置它，并讨论了规划注意事项。它还描述如何准备和维护 DB2 数据库，以便可以搜索文本。

如果您不太熟悉 DB2 Net Search Extender 并且想了解如何使用 DB2 Net Search Extender 函数来搜索文本，则应阅读此部分。

『第二部分 参考』

此部分提供了有关 DB2 Net Search Extender 函数和命令的参考信息。

如果您熟悉 DB2 Net Search Extender 概念和任务，但是需要有关特定 DB2 Net Search Extender 函数或命令的信息，则应阅读此部分。

『第三部分 附录』


此部分提供了 DB2 Net Search Extender 的附加参考信息。它包含有关迁移、内存使用情况、视图、文档模型、消息和代码的信息。


如果需要有关 DB2 Net Search Extender 的特定参考信息，则阅读此部分。


如何阅读语法图

在整本书中使用下面定义的结构描述语法:

- 顺着线路从左到右和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。

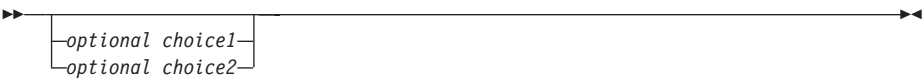


- 若可从两个或多个项目中进行选择，则它们出现在堆栈中。

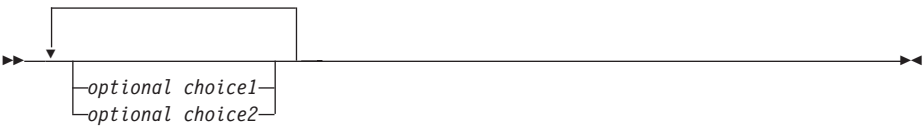
若必须选择其中一个项目，则堆栈的其中一个项目出现在主路径上。



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。



堆栈上方的重复箭头指示可选择堆栈中的多个项目。



- 关键字以大写字母出现；必须严格按显示拼写它们。变量以小写字母出现（例如，srcpath）。它们表示语法中用户提供的名称或值。
- 若显示标点符号、圆括号、算术运算符或其它这样的符号，则必须作为语法的一部分输入它们。

相关信息

IBM DB2 通用数据库版本 8

- *IBM DB2 通用数据库快速入门* 版本 8 DB2 服务器版 (GC09-4836)、DB2 客户机版 (GC09-4832)、DB2 Connect 个人版 (GC09-4834)、DB2 个人版 (GC09-4838) 和 IBM Data Links Manager 版 (GC09-4829-00)。这些书籍描述在适当的平台上如何规划、安装、配置和迁移至 DB2 通用数据库。
- *IBM DB2 Universal Database Administration Guide Version 8 Planning* (SC09-4822)、*Performance* (SC09-4821) 和 *Implementation* (SC09-4820)。这些书籍描述如何设计和实现 DB2 数据库。
- *IBM DB2 Universal Database Call Level Interface Guide and Reference, Volume 1 Version 8* (SC09-4849)。本书描述如何开发使用“DB2 调用层接口”(与 Microsoft ODBC 规范兼容的可调用 SQL 接口)存取 DB2 数据库的应用程序。
- *IBM DB2 Universal Database Call Level Interface Guide and Reference, Volume 2 Version 8* (SC09-4850)。本书描述如何开发使用“DB2 调用层接口”(与 Microsoft ODBC 规范兼容的可调用 SQL 接口)存取 DB2 数据库的应用程序。
- *IBM DB2 Universal Database Command Reference Version 8* (SC09-4828)。本书描述如何使用 DB2 命令行处理器, 并且提供了有关 DB2 命令的参考信息。
- *IBM DB2 Universal Database Replication Guide and Reference Version 8* (SC27-1121)。本书描述如何规划、设置和维护 DB2 数据复制环境。

IBM DB2 通用数据库企业服务器版版本 8

- *IBM DB2 Connect Quick Beginnings for DB2 Connect Enterprise Edition Version 8* (GC09-4833)。本书描述如何在适当平台上规划、安装和配置 DB2 通用数据库扩充企业版。

如何发送意见

您的反馈能帮助 IBM 提供高质量的信息。请将您对本书或其它 DB2 Extender 文档的意见发送给我们。可以通过以下方法来送交您的意见:

- 通过电子邮件将意见发送到 ctscrcf@cn.ibm.com。一定要包括书名、书的部件号、产品的版本和(如果适用的话)您有意见的文本的特定位置(例如, 页号或表号)。

当您发送信息给 IBM 后, 即授予 IBM 非专有权, IBM 可以它认为合适的任何方式使用或分发此信息, 而无须对您承担任何责任。

与 IBM 联系

如果您遇到技术问题, 在与“DB2 客户支持机构”联系之前, 请复查并执行 *Troubleshooting Guide* 中建议的操作。此指南建议您可以收集一些信息以便于“DB2 客户支持机构”能更好地为您服务。

要获取信息或者订购任何 DB2 通用数据库产品，请与您当地 IBM 分部的 IBM 代表联系，或者与任何已授权的 IBM 软件经销商联系。

如果您居住在美国，则可以拨打下列其中一个电话号码：

- 1-800-237-5511 可获得客户支持
- 1-888-426-4343 可了解提供的服务选项

产品信息

如果您居住在美国，则可以拨打下列其中一个电话号码：

- 1-800-IBM-CALL (1-800-426-2255)，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/>

DB2 万维网页面提供了有关新闻、产品描述、教育计划表和其它方面的最新 DB2 信息。

<http://www.ibm.com/software/data/support/>

DB2 支持 Web 页面提供了对常见问题、修订、书籍和最新 DB2 技术信息的访问。

注：本信息可能只提供了英语版。

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

DB2 Extender Web 页面提供了有关当前可用的所有 DB2 Extender 的信息。这些 DB2 Extender 包括 DB2 XML Extender、DB2 Spatial Extender 和 DB2 AIV Extender。

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

DB2 Extender 支持 Web 页面提供了对常见问题、提示与技巧、修订和文档的访问。

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

DB2 Net Search Extender 页面提供有关最新性能技巧的信息。

www.elink.ibm.com/public/applications/publications/cgibin/pbi.cgi

“出版物中心”提供了有关如何订购或下载出版物的信息。

<http://www.ibm.com/certify/index.html>

IBM Web 站点中的 Professional Certification Program 提供了许多 IBM 产品（包括 DB2）的证书测试信息。

在 Compuserve 上: GO IBMDB2

输入此命令以访问 IBM DB2 系列论坛。通过这些论坛支持所有 DB2 产品。

有关在美国以外的国家或地区如何与 IBM 联系的信息，请参考 *IBM Software Support Handbook* 中的 Appendix A。要访问此文档，请访问以下 Web 页面：
<http://techsupport.services.ibm.com/guides/contacts.html>

注：在某些国家或地区中，IBM 授权的经销商应与他们的经销商支持机构而不是“IBM 支持中心”联系。

更改摘要

DB2 Net Search Extender V8.1 F2 引入下列新功能:

- 对分区数据库的支持
- `db2ext.highlight` 函数
- 使用增量索引更新（它使用“DB2 复制”）创建和维护对昵称的文本索引
- 对 Outside-In 过滤软件（由 Stellent 提供）的支持

第 1 部分 用户指南

第 1 章 概述和概念

DB2 Net Search Extender V8.1 是 DB2 Extender™ 系列的其中一个产品。

它替换了 DB2 Text Information Extender V7.2 和 Net Search Extender V7.2，并为用户和应用程序员提供了一种使用 SQL 查询来搜索存储在 DB2、其它数据库和文件系统中的全文本文档的快速、通用和智能的方法。

关键概念

要完全了解 DB2 Net Search Extender 的功能，需要了解关键术语（在本节中是以**粗体**显示的）和提供的各种选项。还需要对“DB2 通用数据库”的概念和术语有一个基本了解。

从本质上来说，DB2 Net Search Extender 搜索数据库表的列中保存的**文本文档**。

文本文档必须是可唯一标识的。为此，Net Search Extender 使用表的**主键**。

文档可以采用各种格式，例如，HTML 或 XML。

Net Search Extender 并不是按顺序搜索所有文本文档（这将花费大量时间），而是创建**文本索引**来使文档可搜索。

文本索引是由从文本文档中抽取的重要**术语**组成的。

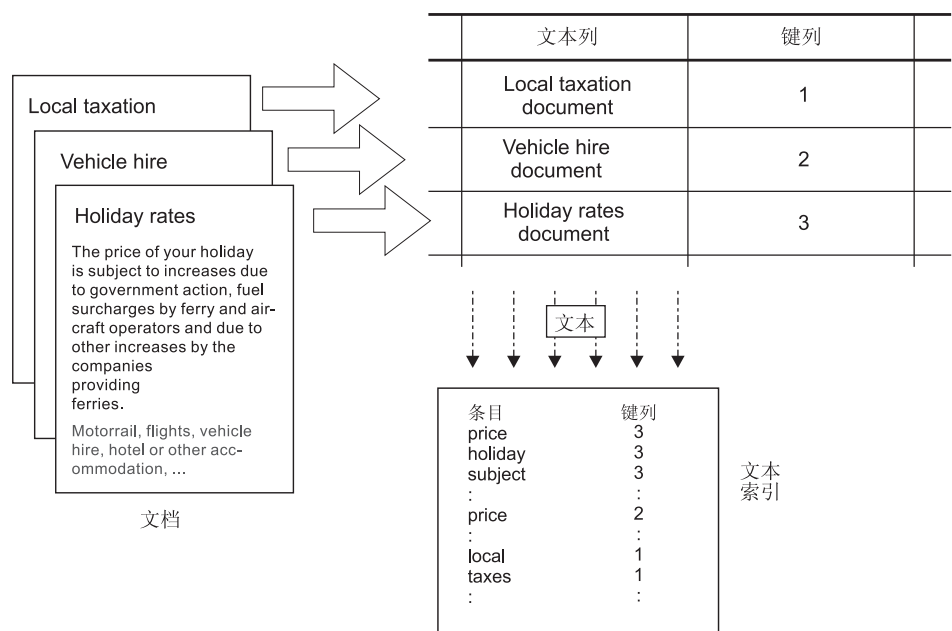


图 1. 创建文本索引

文本索引创建是定义和声明索引属性（例如，索引的位置）的过程。创建之后，文本索引不包含任何数据。**索引更新**是向文本索引中添加数据的过程。第一次索引更新将把文本列中的所有文本文档添加到索引中。它被称为**初始更新**。

当使用文本索引进行搜索时，必须考虑表与文本索引之间的同步问题，这是因为必须在文本索引中反映对表进行的任何后续更改（例如，添加、删除和更新文本文档）。

Net Search Extender 中的同步是根据**触发器**来进行的，触发器会将有关新的、已更改的和已删除的文档信息自动存储在**日志表**中。每个文本索引都有一个日志表。将日志表的内容应用于其相应的文本索引的过程被称为**增量更新**。

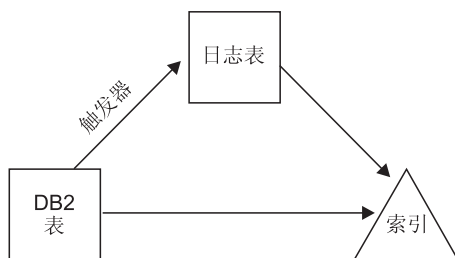


图 2. 增量更新过程

可以使用**手工**或**自动**选项来更新文本索引。自动选项使用更新调度来设置天数和时间。

注意，这些选项**都不会**使更新、删除和插入文本文档的事务范围内的文本索引同步。Net Search Extender 异步建立文本索引提高了性能和并行性。在独立事务内对很小一部分索引的副本应用更新。在用副本代替原始索引的很短时间段内，会锁定索引只允许只读存取。这对搜索操作是不可视的，有关信息，参见第 29 页的第 5 章，『Net Search Extender 实例服务』。

文本索引具有某些属性，例如，索引文件位置和自动更新属性。如有必要，可以更改其中一些属性。这称为**改变索引**。

这样的一种属性是：ORDER BY 短语是否应根据表列对文本索引进行预先排序。在这种情况下，初始更新将按指定的次序对文本文档建立索引，并且按此次序返回搜索结果。

例如，根据书籍价格来指定预先排序的书籍摘要。当查找有关关系数据库系统的最便宜书籍时，可以限制文本搜索只返回前几本书，因为这些书籍是最便宜的。但是，如果没有预先排序的索引，则必须搜索所有书籍，并将这些搜索结果与最便宜的书籍连接起来，这种操作成本会更高。

Net Search Extender 允许每个文本列具有多个预先排序的索引。例如，一个索引用于根据出版日期来对书籍进行预先排序，而第二个索引用于根据书籍价格来对书籍进行预先排序。

通常，创建文本索引之后的第一次更新是初始更新，后续更新是增量更新。但是，当使用预先排序的索引时，您想保持索引次序以防发生更新。这是由**更新时重新创建索引**选项来解决的，每当执行更新时它就会完全重新构建该索引。

更新文本索引之后，可以使用下列一个选项来进行搜索：

- SQL 标量搜索函数
- 存储过程搜索

- SQL 表值函数

由于各个搜索选项具有不同的操作特征，因此，在下列各节中对它们进行了说明。

使用 SQL 标量搜索函数

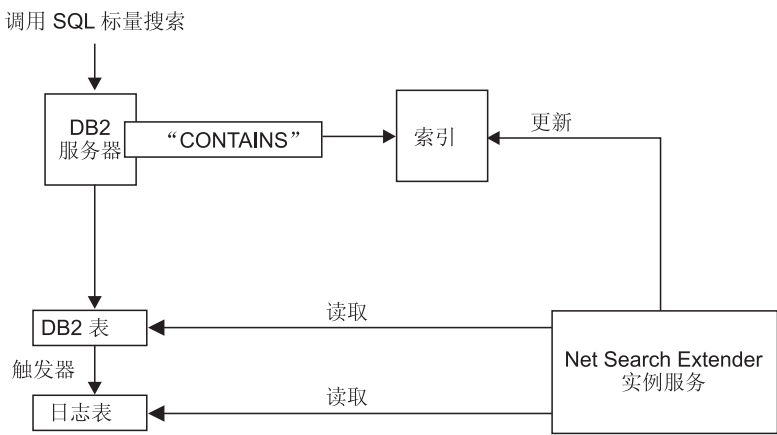


图 3. 使用 SQL 标量搜索函数来进行搜索

Net Search Extender 提供了三个标量文本搜索函数（CONTAINS、NOOFMATCHES 和 SCORE），这些函数无缝地集成在 SQL 中。您可以在 SQL 查询内使用标准 SQL 表达式的相同位置中使用搜索函数。以下是一些典型查询：

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

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

SQL 标量函数将返回一个指示符，它指示文本文档与给定的文本搜索条件的匹配程度。然后，SQL 查询的 SELECT 部分将确定返回给最终用户的信息。

将 SQL 标量搜索函数用作缺省搜索方法。这些搜索函数应该适用于大多数情况，尤其是将文本搜索表达式与其它不同条件进行组合时。

注意：“DB2 优化器”知道有多少个文本文档可能与 CONTAINS 谓词相匹配，以及各种备用存取方案在成本上的差异。优化器将选择成本最低的存取方案。

使用存储过程搜索

存储过程搜索与 SQL 标量搜索函数的工作方式不同。创建文本索引时，必须指定将表或视图外的哪些列返回给最终用户。此数据存储在主存储器中的**高速缓存**中。这将使存储过程搜索能够极快地返回搜索结果。在可以使用高速缓存之前，必须**激活**高速缓存，并且具有相应的**取消激活**命令。

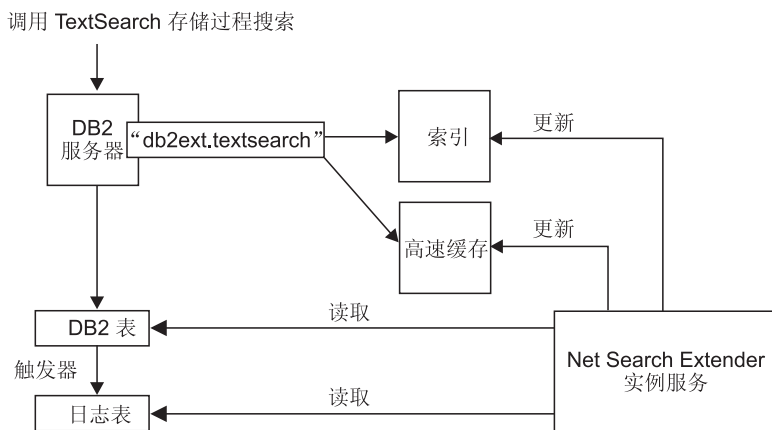


图 4. 使用存储过程搜索

ACTIVATE 命令将数据装入临时高速缓存（它是在激活时从暂存区中创建的）或持久高速缓存（它是在磁盘上维护的）。

使用存储过程来进行搜索时需要计算内存，例如，需要多少内存和应保留多少可用内存来进行索引更新。

存储过程可以处理对视图创建的文本索引。但是，由于不能对视图创建触发器，因此，不能自动识别任何更改。可以将已更改的信息手工添加至日志表，也可以使用 RECREATE 选项。

将存储过程搜索用于对只进行文本搜索的查询感兴趣的高性能 / 高可伸缩性应用程序。例如，不需要将文本搜索结果与其它复杂 SQL 条件的结果进行连接的查询。

存储过程搜索与 SQL 标量搜索函数在功能上的主要区别是：

- 不能在任意 SQL 查询中使用存储过程搜索，它只是对预定义的高速缓存表进行查询。
- 存储过程搜索可以利用对视图创建的索引。
- 存储过程搜索可以利用对某一列创建的已预先排序的多个文本索引。

注意，对于此选项，必须提供大量主内存。有关附加信息，参见第 207 页的附录 B，『使用大量内存』。

使用 SQL 表值函数来进行搜索

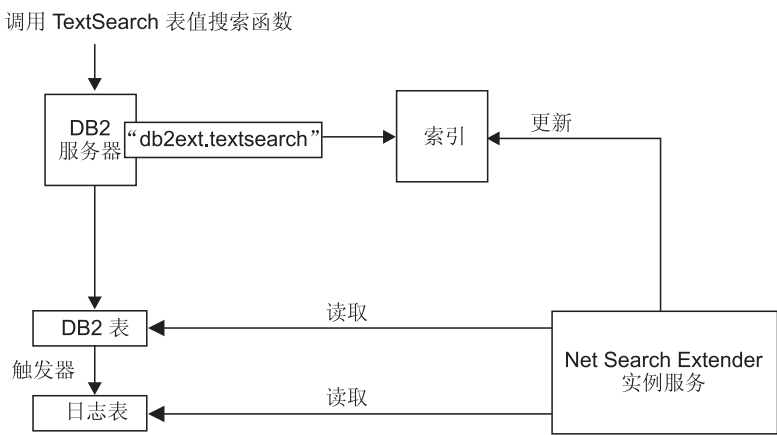


图 5. 使用 SQL 表值函数来进行搜索

SQL 表值函数是 SQL 标量搜索函数与存储过程搜索之间的一种折衷方法。借助 SQL 表值函数，还可以使用 `db2ext.highlight` 函数来获取有关将文档限定为搜索结果的原因的信息。

SQL 表值函数搜索与存储过程搜索在功能上的主要区别是：

- 不需要高速缓存（实际上也没有使用任何高速缓存）。
- 在任何 SQL 语句中都可以使用表值函数。
- 不需要大量主内存。

SQL 表值函数搜索与 SQL 标量搜索函数在功能上的主要区别是：

- SQL 表值函数可以利用有关视图的索引。
- 表值函数可以利用预先排序的文本索引。

在下列情况下使用 SQL 表值函数：通常情况下将使用 SQL 标量函数，但是您想利用有关视图的文本索引或预先排序的文本索引。

附加概念

除了了解 DB2 Net Search Extender 的关键概念之外，还需要说明一些附加概念。

有关开发基于 Net Search Extender 的应用程序的更多信息，参见第 25 页的第 4 章，『规划』。

列变换函数

可以使用您自己的函数来将不受支持的格式或数据类型转换为受支持的格式或数据类型。通过指定“用户定义函数”（UDF），可以使原始文本文档作为输入。UDF 的输出应该是受支持的格式，可以在建立索引期间处理它。

还可以使用此功能对存储在不受支持的外部数据存储中的文档建立索引。在这种情况下，DB2 列包含文档参考，函数将返回具有相关文档参考的文档内容。

参见第 41 页的『对不受支持的数据类型创建文本索引』。

实例服务

“Net Search Extender 实例服务”处理特定于索引的锁定服务和文本索引更新服务（自动和手工）。

有关更多信息，参见第 29 页的第 5 章，『Net Search Extender 实例服务』。

外部存储的数据

在大多数情况下，对其创建文本索引的数据都存储在本机 DB2 表列中（例如，CLOBS 或 VARCHARS 中）。

但是，也支持外部存储的文本文档，例如，存储在文件或其它数据库中的文本文档。对于存储在文件中的文档，DB2 Data Links 功能部件可用。对于存储在其它数据库中的文档，使用 DB2 昵称表来创建文本索引。

参见第 viii 页的『相关信息』。

还可以将列变换函数用于存储在不受支持的外部数据存储中的数据。参见『列变换函数』。

管理视图

在 DB2 Net Search Extender 中提供了几种视图。它们提供了有关文本索引及其属性的信息。

有关信息，参见第 211 页的附录 C，『Net Search Extender 信息目录』。

分区数据库支持

DB2 Net Search Extender 的搜索函数采用下列方法来使用分区数据库支持：

- SQL 标量函数利用对分区表创建的索引。

- 存储过程搜索和 SQL 表值函数只利用分区环境中的一个节点上的表。

对联合数据库中的昵称建立索引

还可以对联合数据库中的昵称创建文本索引，该索引指向远程数据库中的一些表。在这种情况下，日志表的角色（对于增量索引更新）不同于它对于正则表的索引的角色。与正则表不同，不能对昵称创建 DB2 触发器，因此不能使用触发器将有关文档的更改信息插入到日志表中。因此，对昵称的索引的增量更新有两种不同方法：

- 日志表是在联合数据库中以本地方式创建的，应用程序对包含有关昵称的正确更改信息的日志表负责。对于 DB2 视图，这与增量索引更新相似。此选项是缺省选项。
- 已建立“DB2 复制”，以便在通常所说的“更改数据表”（CD 表）（对于 DB2 远程数据库）或者“一致更改数据表”（CCD 表）（对于非 DB2 关系数据库）中捕获对昵称引用的表的更改。于是 DB2 Net Search Extender 可以使用 CD 或 CCD 表而不必为昵称的索引创建日志表。在这种情况下，必须在 DB2TEXT CREATE INDEX 命令中指定捕获表特征。

关键功能

DB2 Net Search Extender V8.1 具有下列关键功能：

- 建立索引
 - 对大量数据快速建立索引
 - 动态更新索引
 - 在建立索引时将表列存储在主内存中，以避免在搜索时执行大量的物理读取操作
 - 通过“DB2 控制中心”选择命令行或接口用于建立索引
 - 不同的文本格式，例如，HTML 和 XML
 - 支持第三方过滤软件“Outside-In”
 - 昵称表支持
 - DB2 Data Link Manager 支持
 - 支持预先排序的文本索引
 - 分区数据库支持
- 搜索
 - 布尔运算
 - 对同一句子或段落中的词语执行近似搜索
 - 对与搜索术语拼写相似的词语进行“模糊”搜索

- 通配符搜索，对整个词语和单个字符使用前面、中间和末尾屏蔽
- 自由文本搜索。对于包含特定文本的文档，搜索自变量是用自然语言表示的
- 突出显示功能，用来显示将特定文档限定为搜索结果的原因
- 同义字支持
- 限制只对文档内的各节进行搜索
- 数字属性支持
- 许多并行用户对大量文本文档进行高速搜索
- 搜索结果
 - 可以在建立索引时指定如何对搜索结果进行排序
 - 可以在搜索大量数据时和期望大型结果列表时指定搜索结果子集
 - 可以使用高匹配项计数对搜索术语设置限制
 - 与“DB2 优化器”组合在一起的内置 SQL 功能将根据期望的搜索结果自动选择最佳方案

DB2 客户机 / 服务器环境中的 DB2 Net Search Extender

DB2 Net Search Extender 搜索功能已集成到 SQL 中并在服务器中执行它。因此，不需要在客户机上安装 Net Search Extender 以发出文本搜索查询。

DB2 Net Search Extender 支持从客户机端对服务器执行管理调用。在客户机端和服务器端安装 DB2 Net Search Extender，或者使用“DB2 控制中心”从客户机端管理 DB2 Net Search Extender。

第 2 章 安装

本章描述如何在 UNIX[®] 和 Windows[®] 系统中安装 DB2 Net Search Extender。

安装之后，运行 DB2 Net Search Extender 安装验证脚本。

系统需求

运行 DB2 Net Search Extender 必须具有下列版本的软件：

- DB2 版本 8.1 修订包 2
- Java 运行时环境 (JRE)。版本取决于 DB2 版本。

DB2 Net Search Extender 可用于下列操作系统：

Solaris 操作环境：

- Solaris 7
- Solaris 8
- Solaris 9

AIX[®]：

- AIX V4.3.3。以下文件集也是必需的：xlC.aix43.rte 5.0.2.x。
- AIX V5.1.0/5.2.0。以下文件集也是获取 64 位代码支持所必需的：xlC.aix50.rte 5.0.2.3 或更高版本。

注意，可以从以下网址下载 AIX 文件集：
<http://techsupport.services.ibm.com/server/fixes>

Windows：

- Windows NT[®] V4
- Windows 2000
- Windows XP

HP：

- HP-UX 11i

Linux：

- 只有 Intel 机器支持 DB2 Net Search Extender。新的 Linux 内核和分发的验证状态频繁更新。要获取受支持的 Linux 软件级别的最新信息，参考 <http://www.ibm.com/software/data/db2/linux/validate>

对于 AIX、Solaris 和 HP-UX，DB2 Net Search Extender 可作为 32 位和 64 位应用程序使用。在 Windows 和 Linux 上，Net Search Extender 只可作为 32 位应用程序使用。

对于所有 DB2 Net Search Extender 操作系统，最低内存需求为 30MB。典型 DB2 Net Search Extender 安装的最小磁盘空间为 50MB。

DB2 Net Search Extender 与“DB2 通用数据库版本 8.1”具有相同的最低软件和硬件需求。有关这些需求以及特定的操作系统补丁程序，参考 *IBM DB2 通用数据库快速入门版本 8 丛书文档*中的相应章节。

任何附加硬件需求都取决于所选择的文本索引的大小和类型。有关 DB2 文档，参见第 viii 页的『相关信息』。

分区 DB2 服务器的安装概述 (UNIX)

确保 DB2 在每个节点上的正确安装和配置。在安装 DB2 之后，需要在每台计算机上安装 DB2 Net Search Extender。

注意

与实例所有者标识不同的受防护用户标识不使用分区数据库。

在 UNIX 上安装

要在 UNIX 上安装，遵循下列步骤：

1. 安装产品。
2. 更新 DB2 实例。

适用于 UNIX 的步骤 1：安装产品组件

要在 UNIX 上安装，遵循下列步骤：

1. 在目标机器上作为 root 用户登录。
2. 切换到您的平台上的正确目录：
 - `cd /<cdrom>`，其中 `<cdrom>` 是 CD-ROM 驱动器路径。
 - `cd /<platform>`

3. 调用 `./nsetup.sh`，并遵循屏幕上显示的指示信息。注意：忽略任何 'Exited with' 消息。

适用于 UNIX 的步骤 2: 更新 DB2 实例

要更新 DB2 实例，遵循下列步骤：

1. 确保您是 root 用户。
2. 针对不同的平台使用下列命令之一：
 - 对于 AIX: `cd /usr/opt/db2_08_01/instance`
 - 对于 Solaris、Linux 和 HP-UX: `cd /opt/IBM/db2/V8.1/instance`
3. 使用 `./db2iupdt <db2instance>` 运行 db2iupdt，其中 `<db2instance>` 是想要与 Net Search Extender 配合使用的现有 DB2 实例用户标识。
4. 注销。

注意

Net Search Extender 自动在 db2icrt 期间创建新的 DB2 实例。

Windows 安装

要在 Windows 上进行安装，必须使用具有管理权限的用户标识登录，然后，遵循下列步骤：

1. 使用 `<cdrom>:\windows\install\setup.exe` 来将文件从程序包传送到目标机器。注意，对于每种 DB2 服务，必须输入用户标识和密码来创建正确的 DB2 Net Search Extender 服务。
2. 在传送数据之后重新引导系统。
3. 调用 `db2text start` 来启动“DB2 Net Search Extender 实例服务”。

注意

每个 DB2 实例都创建一个 Windows 服务。确保 DB2 实例服务在用户帐户下运行，而不是在系统帐户下运行。

目录名和文件名

必须使用 SBCS 字符为所有 Net Search Extender 命令指定目录名和文件名。路径名（包括文件名）的最大长度为 256 个字节或者少于 256 个字节。

安装 Outside-In 库

要将 Net Search Extender 与 Stellent 提供的 Outside-In 软件配合使用，必须为每个平台建立这些库：

- 在 Windows 上，一定要将这些库所在的目录添加至路径环境变量。
- 在 UNIX 上，将所有 Outside-In 库添加至 /opt/IBM/db2/V8.1/lib 目录。

安装验证

在 Windows 和 UNIX 平台上提供了 Net Search Extender 安装验证。

注意

对于分布式数据库，Windows 和 UNIX 平台上的安装验证当前不可用。

Windows 上的安装验证

完成下列步骤以验证是否正确安装了 Net Search Extender。

- 遵循下列步骤来调用 nsesample.bat 管理脚本以设置文本索引：
 1. 调用 db2cmd 以打开 DB2 命令窗口。
 2. 切换到 <sqllib>\samples\db2ext
 3. 从 DB2 命令窗口中调用 nsesample.bat <yourdb>，其中 <yourdb> 是数据库的名称。注意，如果数据库尚不存在，此命令会创建该数据库。
 4. 检查当前目录中生成的输出文件 nsesample.log。
- 然后，调用下列样本查询在 DB2 命令窗口中执行：
 1. 使用 db2 connect to <yourdb> 连接至您的数据库
 2. 使用 db2 -tvf search 执行样本查询
 3. 检查脚本中包含的查询结果。注意，每个查询应当返回一个或多个匹配项。

如果在 nsesample.log 文件中找不到任何错误，并且所有查询都正在工作，则说明成功安装了 Net Search Extender。

UNIX 上的安装验证

完成下列步骤以验证是否正确安装了 Net Search Extender。

- 遵循下列步骤来调用 nsesample 管理脚本以设置文本索引：
 1. 切换到 <instance_owner_home>/sqllib/samples/db2ext

2. 调用 `./nsesample <yourdb>`。注意，如果数据库尚不存在，此命令会创建该数据库。
 3. 检查主目录中生成的输出文件 `nsesample.log`。
- 然后，调用一些样本查询在同一 DB2 命令窗口中执行：
 1. 使用 `db2 connect to <yourdb>` 连接至您的数据库
 2. 使用 `db2 -tvf search` 执行样本查询
 3. 检查脚本中包含的查询结果。注意，每个查询应当返回一个或多个匹配项。

如果在 `nsesample.log` 文件中找不到任何错误，并且所有查询都正在工作，则说明成功安装了 Net Search Extender。

注意

有关迁移信息，参见第 203 页的附录 A，『迁移』。

第 3 章 用户方案

在本章中，通过使用下列逐步执行的示例来了解 Net Search Extender:

SQL 标量搜索示例

此命令行示例演示可用的建立索引和搜索功能。

存储过程示例

此命令行示例使用以上示例中的索引命令。添加了高速缓存后，该示例还演示可用的不同建立索引功能和搜索功能。

SQL 表值函数示例

有关使用 Net Search Extender 的更多信息，参见下列各章:

- 第 25 页的第 4 章，『规划』
- 第 35 页的第 6 章，『创建和维护文本索引』
- 第 83 页的第 8 章，『搜索』

注意

在使用这些示例之前，通过使用安装验证过程来确保成功安装了 Net Search Extender。

使用 SQL 标量搜索函数的简单示例

在 DB2 Net Search Extender 示例中使用下列步骤:

1. 创建数据库
2. 启用数据库以进行文本搜索
3. 创建表
4. 创建全文索引
5. 装入样本数据
6. 使文本索引同步
7. 搜索文本索引

可以在操作系统的命令行上使用现有数据库发出样本命令。对于下列示例，数据库名为 sample。

创建数据库

可以使用以下命令在 DB2 中创建数据库:

```
db2 "create database sample"
```

启用数据库以进行文本搜索

可以象在操作系统的命令行上发出 DB2 命令那样发出 DB2 Net Search Extender 命令。例如, 使用以下命令来启动 “Net Search Extender 实例服务”:

```
db2text "START"
```

使用以下命令来准备数据库, 使它与 DB2 Net Search Extender 配合使用:

```
db2text "ENABLE DATABASE FOR TEXT CONNECT TO sample"
```

对每个数据库只需执行一次此步骤。

创建表

```
db2 "CREATE TABLE books (isbn VARCHAR(18) not null PRIMARY KEY,  
author VARCHAR(30), story LONG VARCHAR, year INTEGER)"
```

此 DB2 命令将创建一个称为 books 的表。它包含 author 列、story 列、isbn number 列和出版书籍的 year 列。

创建全文本索引

```
db2text "CREATE INDEX db2ext.myTextIndex FOR TEXT ON books (story)  
CONNECT TO sample"
```

此命令将为 story 列创建全文本索引。该文本索引的名称为 db2ext.myTextIndex

装入样本数据

```
db2 "INSERT INTO books VALUES ('0-13-086755-1','John', 'A man was  
running down the street.',2001)"  
db2 "INSERT INTO books VALUES ('0-13-086755-2','Mike', 'The cat hunts  
some mice.', 2000)"  
db2 "INSERT INTO books VALUES ('0-13-086755-3','Peter', 'Some men  
were standing beside the table.',1999)"
```

这些命令将把这些书籍的 isbn、作者、情节和出版年份装入表中。

使文本索引同步

要使用 sample 表中的数据来更新文本索引, 使用以下命令:

```
db2text "UPDATE INDEX db2ext.myTextIndex FOR TEXT CONNECT TO sample"
```

搜索文本索引

要搜索文本索引, 使用以下 CONTAINS 标量搜索函数:

```
db2 "SELECT author, story FROM books WHERE CONTAINS
      (story, '\"cat\"') = 1 AND YEAR >= 2000"
```

注意

根据正在使用的操作系统外壳程序的不同，在将文本搜索短语引起来的双引号前面可能需要添加不同的转义字符。以上示例将 "\" 用作转义字符。

此查询将搜索 2000 年或 2000 年以后关于词语 cat 的所有书籍。查询返回以下结果表：

AUTHOR	STORY
Mike	The cat hunts some mice.

受支持的其它函数包括 SCORE 和 NUMBEROFMATCHES。SCORE 将返回一个指示符，指示搜索自变量描述找到的文档的仔细程度。NUMBEROFMATCHES 将返回在获得的文档中找到的查询术语匹配项的数量。

使用高速缓存用法和存储过程搜索的简单示例

在 DB2 Net Search Extender 存储过程搜索示例中使用下列步骤：

1. 使用高速缓存选项创建文本索引。
2. 使索引同步并激活高速缓存。
3. 使用 TEXTSEARCH 存储过程进行搜索。

注意

存储过程示例假定前一示例中的步骤是完整的，并且仍然启用了数据库。

使用高速缓存选项创建文本索引

由于已经启用了数据库，因此，使用以下命令来创建全文本索引：

```
db2text "CREATE INDEX db2ext.mySTPTextIndex FOR TEXT ON books (story)
        CACHE TABLE (author, story) MAXIMUM CACHE SIZE 1
        CONNECT TO sample"
```

在此示例中，全文本索引是针对 story 列的，并且它对 author 和 story 列指定高速缓存表。文本索引的名称为 mySTPTextIndex。

使索引同步并激活高速缓存

要根据插入到表中的数据来更新索引，使用以下命令：

```
db2text "UPDATE INDEX db2ext.mySTPTextIndex FOR TEXT CONNECT TO sample"
```

要激活高速缓存，使用以下命令：

```
db2text "ACTIVATE CACHE FOR INDEX db2ext.mySTPTextIndex FOR TEXT  
CONNECT TO sample"
```

使用“TEXTSEARCH 存储过程”进行搜索

只有在某些情况下才能使用 DB2 Net Search Extender 存储过程。有关详细信息，参见第 7 页的『使用存储过程搜索』。

```
db2 "call db2ext.textSearch  
      ('\cat\'','DB2EXT','MYSTPTEXTINDEX',0,2,0,0,?,?)"
```

此查询将搜索有关 cat 的所有书籍，但是，将只返回前两个结果。在这种情况下，查询将返回以下结果表：

```
Value of output parameters  
-----  
Parameter Name : SEARCHTERMCOUNTS  
Parameter Value : 1  
Parameter Name : TOTALNUMBEROFRESULTS  
Parameter Value : 1
```

```
AUTHOR    STORY  
Mike      The cat hunts some mice.
```

```
Return Status = 0
```

有关搜索语法的更多样本，请查看以下文件：

```
sqlllib\sample\db2ext\search
```

有关查询中使用的其它参数的详细信息，参见第 91 页的『使用存储过程搜索来搜索文本』。

使用 SQL 表值函数的简单示例

可以对在前面示例中创建的文本索引使用 SQL 表值函数。

SQL 表值函数查询对应于先前使用的 CONTAINS 查询。有关信息，参见第 20 页上的“使文本索引同步”。

```
db2 "SELECT author, story FROM books b, table (db2ext.textsearch  
      ('\cat\'','DB2EXT','MYTEXTINDEX', 0, 2, CAST  
      (NULL AS VARCHAR(18)))) T where T.primKey = b.isbn"
```

有关更多详细信息，参见第 8 页的『使用 SQL 表值函数来进行搜索』。

注意

CAST (NULL AS VARCHAR(18)) 对表 books 的主键调用相应的表值函数。

对于每个主键类型，都有一个表值函数。它将为 DB2 标识正确的表值函数。

第 4 章 规划

要以最有效的方式使用 DB2 Net Search Extender，必须在开发之前进行一些规划。这种开发可能涉及到几个小组，包括数据库管理、接口和系统设计人员、系统设计师和开发者。

下列各节对应该考虑的各个方面提供了指导。

有关开发基于 DB2 Net Search Extender 的应用程序的更多信息，参见下列各章：

- 第 29 页的第 5 章，『Net Search Extender 实例服务』
- 第 35 页的第 6 章，『创建和维护文本索引』
- 第 83 页的第 8 章，『搜索』

目录位置和索引存储器

需要用于索引的磁盘空间取决于要对其建立索引的数据的大小和类型。指导原则是，对单字节文档建立索引，需要保留大约要对其建立索引的文档大小的 0.7 倍的磁盘空间。对于双字节文档，保留与想要对其建立索引的文档的大小相同的磁盘空间。

工作目录中的临时文件需要的空间量是索引目录中的最终索引文件需要的空间量的 1.0 到 4.0 倍。

如果具有几个大型索引，则应将它们存储在独立的磁盘设备上，尤其是在索引更新或搜索期间并行存取这些索引的情况下更应如此。

还必须指定将存储文本索引的目录。确保具有足够的磁盘空间，并且 DB2 实例所有者对该目录具有写存取权。

文档格式和受支持的代码页

DB2 Net Search Extender 需要知道您打算搜索的文本文档的格式（或类型）。此信息对于为文本文档建立索引是必需的。

DB2 Net Search Extender 支持下列文档格式：

TEXT	明文（例如，平面 ASCII）
HTML	超文本标记语言

XML 扩展标记语言

GPP 通用格式（带有用户定义标记的平面文本）

Outside-In (INSO)

过滤软件以便从 PDF 和其它常见文本格式化工具（例如，Microsoft Word）中抽取文本内容。有关进一步的信息，参见『Outside-In 过滤软件』。

对于文档格式 HTML、XML、GPP 和 Outside-In 过滤器格式，可以限制只搜索文档的特定部分。第 97 页的第 9 章，『使用结构化文档』说明如何定义和使用文档模型。

其中 Outside-In 过滤器不能用于不受支持的文档格式，您可以编写“用户定义函数”（UDF）。必须在创建索引时指定此 UDF 并将数据从不受支持的格式转换为受支持的格式。

有关更多信息，参见第 130 页的『CREATE INDEX』。

如果文档采用其中一种受支持的“编码字符集标识符”（CCSID），则可以对文档建立索引。这些 CCSID 也称为代码页。有关这些代码页的列表，参见第 221 页的附录 D，『受支持的 CCSID』。

要检查数据库代码页，使用以下 DB2 命令：

```
db2 GET DB CFG for <dbname>
```

为了保持一致，DB2 通常将文档的代码页转换为数据库的代码页。但是，当将 DB2 数据库中的数据存储在具有二进制数据类型（例如，BLOB、FOR BIT DATA 或 datalink value）的列中时，DB2 不会转换数据，且文档将保持其原始 CCSID。

注意，如果具有两种不同的代码页，则在创建文本索引或进行搜索时可能会导致问题。有关进一步的信息，参见第 41 页的『对二进制数据类型创建文本索引』。

Outside-In 过滤软件

DB2 Net Search Extender 支持第三方文档过滤软件。例如，Stellent 的 Outside-In，可以使用该软件来从 PDF 文件或者在不使用本机应用程序的情况下从以常见文本格式工具的专有格式编写的文档中抽取文本内容。示例格式包括 Microsoft Word 和 Lotus Word Pro。

在 UPDATE INDEX 运行期间，将 Outside-In 库作为插件装入。这些库不是 Net Search Extender 的一部分，需要单独安装。需要确保 Net Search Extender 可以找到 Outside-In 库。有关设置和使用这些库的信息，参见第 16 页的『安装 Outside-In 库』。

Outside-In 软件不仅可以生成文本内容，还可以生成结构信息（例如，字段）。Net Search Extender 还可以定制要将 Outside-In 生成的文档信息的哪一部分存储在索引中。为此，需要应用特定类型的文档模型（Outside-In 文档模型）。

有关信息，参见第 195 页的『为 Outside-In 过滤的文档定义文档模型』。

Stellent Web 站点为 <http://www.stellent.com>。

要查看过滤格式的列表，使用以下 URL:

```
http://www.stellent.com/intradoc-cgi/nph-idc.cgi.exe/p31019225.pdf?IdcService=
GET_FILE&noSaveAs=1&Rendition=Web&RevisionSelectionMethod=
LatestReleased&allowInterrupt=1&dDocName=p31019225
```

注意

Outside-In 过滤软件只可用于 32 位实例。没有提供 64 位支持。

User 角色

以下是 Net Search Extender 用户的不同角色和权限:

DB2 实例所有者

DB2 实例所有者用户可以启动和停止 DB2 Net Search Extender 的实例服务并控制锁定服务。此外，DB2 实例用户成为每个已启用的数据库的 DBADM。这使得由 DB2 Net Search Extender 来驱动所有数据库更改的中央控制点。

必需的 DB2 权限

对 ENABLE DATABASE 授予了 DBADM 权限。

必需的文件系统权限

对所有文本索引目录的读写存取权以及对模型文件的读存取权。

实例所有者的命令

DB2TEXT START、DB2TEXT STOP 和 DB2TEXT CONTROL。

这些命令只在服务器上允许的。在分布式 DB2 环境中可以是任何服务器。每个命令检查运行该命令的用户是不是 DB2 实例所有者。注意，就权限或性能方面来说，在 UNIX 系统上使用独立的受防护用户标识不会影响 Net Search Extender 处理。

数据库管理员

数据库管理员可以启用和禁用数据库与 DB2 Net Search Extender 配合使用。

必需的 DB2 权限

DBADM (对于 ENABLE DATABASE 的 SYSADM 权限)。

数据库管理员的命令

DB2TEXT ENABLE DATABASE 和 DB2TEXT DISABLE DATABASE。

文本表所有者

文本表所有者可以创建、删除和更改索引。注意，他们必须能够控制索引的位置和对全文索引的更新。

必需的 DB2 权限和特权

文本表的所有者。

文本表所有者的命令:

DB2TEXT CREATE INDEX、DB2TEXT DROP INDEX、DB2TEXT ALTER INDEX、DB2TEXT ACTIVATE CACHE、DB2TEXT DEACTIVATE CACHE、DB2TEXT UPDATE INDEX、DB2TEXT CLEAR EVENTS 和 DB2EXTTH。

注意，命令实现在 DB2 实例所有者的用户标识下部分运行。因此，在创建或改变文本索引之前，为实例所有者授予必需的文件系统存取权。

使用命令行或“DB2 控制中心”界面来建立索引

要建立索引，可以使用命令行选项，也可以使用“DB2 控制中心”界面。

第 5 章 Net Search Extender 实例服务

“DB2 Net Search Extender 实例服务” 包含下列服务：

- 锁定服务
- 更新服务

本章说明如何启动和停止 “DB2 Net Search Extender 实例服务”。它还详细讨论了 “锁定服务” 和 “更新服务”。

启动和停止 DB2 Net Search Extender

在可以创建文本索引和搜索文档之前，必须启动 “DB2 Net Search Extender 实例服务”。

要启动 “实例服务”，登录到 DB2 实例所有者用户标识（仅适用于 UNIX 系统），并输入以下命令：

```
db2text start
```

要停止 “实例服务”，输入以下命令：

```
db2text stop
```

注意，每个 DB2 实例必须具有一个 “Net Search Extender 实例服务”。锁定服务为多个数据库维护锁定。

锁定服务

当启动 DB2 Net Search Extender 时，将自动启动锁定服务。锁定服务需要对 Net Search Extender 中的文本索引的并行存取同步。

锁定服务可以确保没有两个进程同时尝试更改一个文本索引，或者确保当一个进程正在更改一个文本索引数据时没有其它进程读取该文本索引数据。因此，大多数进程在启动之前将请求锁定文本索引，而在完成处理之后再次释放锁定。

注意，一定不能将 Net Search Extender 文本索引的锁定服务与用来控制对 DB2 表的存取的 DB2 锁定相混淆。

使用锁定服务

在 Net Search Extender 中，有几种不同类型的锁定来控制对索引的并行存取。不同的锁定取决于在搜索请求的情况下是否只读取文本索引，或者是否需要计算对文本索引的更改，然后将其写入文件中。

在 `db2text start` 运行期间，将自动启动锁定服务。对文本索引具有下列类型的锁定：

S 锁定

用于共享只读存取。例如，搜索请求。

U 锁定

用于当使用并行读取存取来计算对索引的更改（更新）时进行读写存取。例如，搜索请求。

X 锁定

用于较短时间内进行互斥读 / 写存取，在这段时间内将把更改真正写入索引中。

IX 锁定

用于当更新进程正在等待 X 锁定时防止进行任何新的 S 锁定的期望互斥读 / 写存取。

每个 DB2 实例都具有一个 Net Search Extender 锁定服务。锁定服务为多个数据库维护锁定。

锁定服务配置文件为 `db2extlm.cfg`。它存储在 `<instance_owner_home>/sqlllib/db2ext` 上（对于 UNIX 系统）和 `<sqlllib>\<DB2INSTANCE>\db2ext` 上（对于 Windows）。

仅当“Net Search Extender 实例服务”是在 `db2text start` 期间启动时，对配置文件所做的更改才会生效。有关进一步的信息，参见第 108 页的『CONTROL』。用户可以设置下列值：

- 数据库的最大数目
- 每个数据库的最大索引数
- 每个索引允许的最大锁定数（并行用户）
- 获得锁定的等待时间和尝试次数

配置文件的缺省值为如下所示：

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

```

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

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

latchTimeout = "80"

/>

```

语法为 <default attribute=value.../>, 属性的含义如下:

maxDbcs

锁定服务可以处理的数据库数目 (大于 1 的整数)。

maxIdxPerDb

可以锁定的每个数据库的索引数 (大于 1 的整数)。对于所有数据库, 此值都是相同的。

maxLocksPerIdx

一个索引中可以同时存在的锁定数 (大于 1 的整数)。对于所有索引, 此值都是相同的。

sWait/sAttempt

当请求 S 锁定时, sAttempt 是指当不能立即授予锁定时尝试的次数。sWait 是这些尝试之间的等待时间 (大于 1 的整数)。这些参数也适用于 IX 锁定。

uWait/uAttempt

当请求 U 锁定时, uAttempt 是指不能立即授予锁定时尝试的次数。uWait 是这些尝试之间的等待时间 (大于 1 的整数)。

xWait/xAttempt

当请求 X 锁定时, xAttempt 是指不能立即授予锁定时的尝试次数。xWait 是这些尝试之间的等待时间 (大于 1 的整数)。

latchTimeout

这是时间间隔锁定服务的附加等待时间。要确定锁定的总等待时间, 使用以下计算公式:

$$\text{waiting time} = \# \text{ attempts} * (\# \text{ waits} + (2 * \# \text{ latchTimeout}))$$

等待时间是以毫秒为单位计算的。注意, 对于每次尝试, latchTimeout 值在添加到总的等待时间时都要加倍。

查看锁定快照

可以使用下列命令之一来查看锁定快照:

- 对于单个文本索引:

```
db2text CONTROL LIST ALL LOCKS FOR DATABASE mydatabase INDEX myindex
```

- 对于数据库的所有锁定文本索引:

```
db2text CONTROL LIST ALL LOCKS FOR DATABASE mydatabase
```

注意, 只有实际锁定的索引在列表中。

当第一次锁定文本索引时, 就为数据库和锁定服务中的文本索引保留了内存。如果锁定了更多文本索引, 则在锁定服务中还要为这些索引分配内存。仅当删除文本索引或禁用数据库或者重新启动 Net Search Extender 服务时才会释放此内存。这就意味着文本索引或数据库在锁定服务中会消耗内存 (即使当前没有设置任何锁定)。

命令 `db2text CONTROL CLEAR ALL LOCKS` 将强制释放对数据库或索引的所有锁定。有关如何使用此命令的详细信息, 参见第 108 页的『CONTROL』。注意, 此命令不会释放为数据库或索引分配的任何内存。要释放内存, 必须删除该索引或禁用数据库, 或者重新启动 Net Search Extender 服务。

更新服务

更新服务是在 `db2text start` 运行期间启动的。在指定时间, 这些服务自动更新文本索引。注意, 文本索引不会立即与用户表同步。

如果使用以下命令需要更新索引, 则在创建索引期间可以指定“更新服务”的检查频率:

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

在此示例中, 这意味着每个星期一和星期三的晚上 12 点和上午 12 点唤醒“更新服务”, 并检查是否要对索引 `db2ext.title` 完成某些工作。注意, 在可以启动自动索引更新以便与数据库同步之前至少需要五个更改。

有关参数的更多详细信息, 参见第 130 页的『CREATE INDEX』。

在分区数据库环境中, 更新服务只在一个节点上启动。

注意

每分钟使用 UPDATE FREQUENCY 将给您的机器带来很高的工作负荷。要避免这种情况，只将 UPDATE FREQUENCY 用于极其有限数目的索引。

第 6 章 创建和维护文本索引

本章提供了有关创建和维护文本索引的信息，它包括下列方面的内容：

- 介绍 `db2text` 命令
- 启用一个数据库以进行文本搜索
- 安装 Outside-In 库
- 创建不同数据类型的文本索引
- 使用增量索引更新（它使用“DB2 复制”）来对昵称创建文本索引
- 创建存储过程可以使用的文本索引
- 有关视图的文本索引
- 维护索引

还具有有关避免可能发生的代码页问题的信息，以及可能需要注意的性能注意事项。

在创建文本索引之前，确保已经考虑了第 25 页的第 4 章，『规划』中的先决条件。其它建立索引先决条件包括使用 `db2text start` 命令来启动“DB2 Net Search Extender 实例服务”。

有关创建文本索引和使文本可搜索的示例，参见第 19 页的第 3 章，『用户方案』。

注意

还可以使用“DB2 控制中心”来创建和维护文本索引。参见第 57 页的第 7 章，『使用 DB2 控制中心』。

介绍 `db2text` 命令

以下是 DB2 Net Search Extender 命令的一个示例：

```
db2text ENABLE DATABASE FOR TEXT
```

注意

db2text 命令，例如，db2text ENABLE DATABASE FOR TEXT 和 db2text CREATE INDEX 也称为命令。

提示

对于每个创建和索引维护命令，可以指定数据库、用户和密码。

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

注意，如果在 db2text 命令中忽略了连接选项，则 DB2DBDFT 环境变量将指定数据库。

要显示命令列表，输入以下命令：

```
db2text ?
```

要显示个别命令的语法，输入以下命令：

```
db2text ? command
```

例如，要显示 CREATE INDEX 命令的语法，使用以下命令：

```
db2text ? CREATE INDEX
```

如果已经成功处理了该命令，则 db2text 将返回 0，如果尚未处理该命令，则将返回 1。注意，如果存在文档错误，但是索引仍然要更新，则 db2text 命令将返回 0。

注意

系统外壳程序将解释诸如 ?、(、)、*、! 和 " 之类的特殊字符。因此，如果命令中包含这些字符，则应使用引号或转义字符。

以下是使用特殊字符的 UNIX 命令的一个示例：

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

启用数据库

总结

何时启用数据库

对包含要搜索的文本列的每个数据库启用一次。

命令

ENABLE DATABASE FOR TEXT

权限

SYSADM

此命令准备已连接的数据库供 DB2 Net Search Extender 使用。

此命令还声明在第 167 页的第 15 章，『SQL 标量搜索函数和 SQL 表值函数』中描述的 DB2 Net Search Extender 搜索函数和过程。

当启用数据库时，创建下列表和视图：

db2ext.dbdefaults

显示索引、文本和处理特征的数据库缺省值。

db2ext.textindexformats

显示受支持格式的列表和使用的模型文件。

db2ext.indexconfiguration

显示索引配置参数。

db2ext.textindexes

用来跟踪所有文本索引的目录视图。

db2ext.proxyinformation

显示关于使用代理服务器存取文件的代理信息。

有关所有视图的信息，参见第 211 页的附录 C，『Net Search Extender 信息目录』。

当启用了数据库时，它将保持处于启用状态，直到您禁用它为止。

注意

与实例所有者标识不同的受防护用户标识不使用分区数据库。

禁用数据库

总结

何时禁用数据库

当您不再打算在此数据库中进行文本搜索时。

命令 `DISABLE DATABASE FOR TEXT`

权限 数据库的 DBADM

当 DB2 Net Search Extender 准备数据库以供使用时，将进行某些管理更改。本节描述可帮助您取消此过程的一些功能。

要禁用已连接的数据库，使用以下命令：

```
db2text DISABLE DATABASE FOR TEXT
```

当禁用数据库时，还要删除下列对象：

- 在启用数据库时已创建的 DB2 Net Search Extender 目录视图。
- DB2 Net Search Extender 的 SQL 函数（UDF）的声明。

要禁用数据库并除去所有文本索引，使用以下命令：

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

注意

如果在数据库中定义了任何文本索引，则禁用数据库将失败。建议依次除去这些索引，然后检查是否发生任何问题。如果将禁用数据库用于文本强制命令，则它只保证除去数据库中的 Net Search Extender 目录表。

但是，如果一些索引未能完全删除，则可能仍然有一些资源需要手工清除。这些资源包括：

- 索引、工作和高速缓存目录中的文件
- ctedem.dat 中的调度程序条目
- 其中索引是使用复制捕获选项创建的，而远程数据库的表中的 IBMSNAP_SIGNAL、IBMSNAP_PRUNE_SET 和 IBMSNAP_PRUNCNTL 条目必须手工删除。使用 APPLY_QUAL="NSE" || <instance name> and TARGET_SERVER= <database name> 命令很容易标识这些条目。

在以下示例中，实例是 DB2，而数据库是 SAMPLE。

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

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

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

创建文本索引

总结

何时删除文本索引	
对包含要搜索的文本的每个列创建一次。	
命令	CREATE INDEX ... FOR TEXT ...（参见下列示例）
权限	表的 CONTROL 权限

尽管对下列数据类型具有不同的要求，您还是可以对受支持的数据类型创建文本索引：

创建文本索引

- 二进制数据类型
- 不受支持的数据类型
- DATALINK 数据类型

为存储过程搜索创建文本索引也具有不同的需求。

当创建文本索引时，还要创建下列对象：

日志表 它将跟踪用户表中已更改的所有行。注意，如果选择**更新时重新创建索引**选项或者使用复制捕获表，则不会创建日志表。

事件表 此表在更新文本索引期间将收集有关问题的信息。

用户表上的触发器

每当添加、删除或更改列中的文档时，这些触发器就会将信息添加至日志表。当下一次建立索引时，该信息对于索引同步是必需的。

注意，仅当创建日志表时才会创建触发器，并且文本索引是在基本表而不是视图或昵称表上创建的。

要优化性能和磁盘空间，使用 `CREATE INDEX` 命令来为表指定另一个表空间。

注意

使用 `DB2 LOAD` 命令来导入文档可能会导致问题，这是因为不会触发触发器，并且不可能对已装入的文档建立增量索引。

因此，最好是在 `DB2 IMPORT` 命令激活触发器时才使用该命令。

以下示例将对 `htmltab` 表中的 `HTMLFILE` 文本列创建文本索引。

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

此表上的主键是必需的。

创建索引的缺省值来自 `db2ext.dbdefaults` 视图。

如果在建立索引期间发生了错误，则会将通常所说的**索引更新事件**添加至事件表。例如，找不到为建立索引而排队的文档。有关附加信息，参见第 217 页的『事件视图』。

要取消由 `CREATE INDEX` 所做的更改，使用 `DROP INDEX` 命令。有关此信息，参见第 54 页的『删除文本索引』。

要使文本索引与数据库同步，使用以下命令：

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

注意，只有在同步之后才能查找文档。

搜索总结

根据在创建索引期间所选择的选项，可以采用不同的搜索方法：

- SQL 标量搜索函数处理所有文本索引（除了对视图创建的文本索引）。
- 存储过程搜索函数只处理使用高速缓存创建的文本索引。
- SQL 表值函数处理所有文本索引（包括对视图创建的文本索引）。

对二进制数据类型创建文本索引

当将数据存储在具有二进制数据类型（例如 BLOB、FOR BIT DATA 或 datalink 值）的列中时，DB2 不会转换数据。这意味着文档将保留它们的原始代码页（CCSID），这在创建文本索引时将导致问题，这是因为您可能具有两个不同的代码页。因此，需要确定您正在使用数据库的代码页，还是正在使用在 CREATE INDEX 命令中指定的代码页。

为了避免产生此问题，应在创建文本索引时指定代码页：

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

如果没有指定代码页，则通过调用以下命令来检查已经使用了哪个 CCSID 来创建索引：

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

注意，不支持将具有不同代码页的文档放在一个文本索引中。有关 DB2 如何转换文档代码页设置的信息，请参考《DB2 通用数据库管理指南》。参见第 viii 页的『相关信息』。

注意，对字符数据类型创建索引时不存在这种问题。

对不受支持的数据类型创建文本索引

要创建索引，文本列必须属于下列数据类型之一：

- CHAR
- VARCHAR
- LONG VARCHAR

创建文本索引

- CLOB
- GRAPHIC
- VARGRAPHIC
- LONG VARGRAPHIC
- DBCLOB
- BLOB
- DATALINK

如果文档在其它类型（例如，用户定义类型（UDT））的列中，则必须提供将用户类型作为输入的函数，和作为输出类型来提供上面提到的一种类型。

指定此变换函数的名称。有关进一步的信息，参见第 130 页的『CREATE INDEX』。

示例：您打算将压缩文本存储在表中。

1. 在交互式 SQL 会话中为文本创建用户定义类型（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 UDTTABLE (author VARCHAR(50) not null,  
                               text COMPRESSED_TEXT, primary key (author))"  
db2 "INSERT ..."
```

3. 例如，创建一个称为 `uncompress` 的用户定义函数（UDF）。这将接收到一个类型为 `COMPRESSED_TEXT` 的值，并且返回相应的未压缩文本，例如，值 `CLOB(10M)`。

4. 按以下方法创建文本索引来指定 `uncompress` UDF：

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

对 DATALINK 数据类型创建文本索引

DB2 Net Search Extender 支持 DATALINK 数据类型。

1. 如果正在使用代理，则将一行添加至 `db2ext.proxyinformation` 表，对主机名、超时值（以秒计）和端口“proxy”或“socks”进行计数。

```
db2 INSERT into db2ext.proxyinformation values  
    ('hostname', '80' 'proxy', 10)
```

注意，此表中只允许存在一行。触发器会保证这一点。

2. 有关如何在不同平台上（例如，在“DB2 信息中心”中）设置 Java 环境的详细信息，参见“DB2 通用数据库版本 8”文档。实质上，必须调整数据库管理器配置。

Data Link UDF 将返回 BLOB（100 KB）。要更改返回大小，使用 DB2EXTDL 并更新数据库管理器配置参数 `java_heap_sz`。

安装 Data Links jar 文件

借助 Net Search Extender，可以对存储在使用 DB2 Data Links 功能部件引用的文件中的数据建立索引。为此，需要安装 Data Links jar 文件 `ctedludf.jar`。在 DB2 命令行处理器中，运行以下命令：

- 对于 UNIX:

```
call sqlj.install_jar
('file: /<instance_owner_home>/sqllib/java/ctedludf.jar','ctedludf_jar')
```

- 对于 Windows:

```
call sqlj.install_jar
('file:D:\sqllib\java\ctedludf.jar','ctedludf_jar')
```

还需要使用以下命令更新 Java 堆大小：

```
db2 update dbm cfg using JAVA_HEAP_SZ 1024
```

使用以下命令注销所有平台上的 jar 文件：

```
call sqlj.remove_jar('ctedludf_jar')
```

有关错误消息的列表，参见第 273 页的附录 K，『Data Link 消息』。

使用增量索引更新（它使用“DB2 复制”）来对昵称创建文本索引

在使用复制捕获表对昵称创建文本索引之前，必须执行下列步骤：

注意

下列步骤只是对该过程提供了概述，并不是一个示例。

1. 使用所有服务器定义和包装器定义设置 DB2 联合数据库。
2. 在远程服务器中设置复制控制表和 Capture 程序。这就是昵称的源表驻留的地方。参见 *DB2 Replication Guide and Reference, Version 8* 的第 2 章“Setting up for Replication”。如果 DB2 不自动创建昵称，则必须使用下列表的一个模式名以在联合 DB2 数据库中创建昵称：
 - IBMSNAP_SIGNAL
 - IBMSNAP_PRUNE_SET

- IBMSNAP_PRUNCNTL
- IBMSNAP_REGISTER
- IBMSNAP_REG_SYNC（仅适用于非 DB2 远程源）

在执行此步骤之后，复制控制表的昵称是作为 DB2 联合数据库上的一种“捕获控制模式”中的昵称提供的。此模式名对于 DB2TEXT CREATE INDEX 命令很重要。

3. 将表注册为复制源。有关详细信息，参见 *DB2 Replication Guide and Reference, Version 8* 的第 3 章“Registering tables and views as replication sources”。有关注册要对其创建索引的昵称的限制，参见第 136 页以获取信息。
4. 如果在注册步骤中 DB2 未自动创建昵称，则在联合数据库中为复制捕获表创建昵称。复制捕获表可以是“更改数据”（CD）表，也可以是“一致更改数据”（CCD）表。此昵称是 DB2TEXT CREATE INDEX 命令的一个参数。

注意，一定不能更改列名

IBMSNAP_OPERATION、IBMSNAP_COMMITSEQ、IBMSNAP_INTENTSEQ 和主键列的名称。

5. 如果正在使用 DB2 复制源，则确保 Capture 程序正在运行。强烈建议不要对 Capture 程序使用冷启动。如果使用冷启动，则必须重新插入 APPLY_QUAL LIKE 'NSE%' 的 IBMSNAP_SIGNAL 表中的所有行。在以下 SQL 语句中，可以看到这是如何完成的：

```
INSERT INTO <capture control schema>.IBMSNAP_SIGNAL
SELECT CURRENT_TIMESTAMP, 'CMD', 'CAPSTART', MAP_ID, 'P'
FROM <capture control schema>.IBMSNAP_PRUNCNTL
WHERE APPLY_QUAL LIKE 'NSE%';
```

6. 可以使用以下示例来通过使用复制对昵称创建文本索引：

```
DB2TEXT
CREATE INDEX <indexname> FOR TEXT ON <nickname> (< text column>)
REPLICATION CAPTURE TABLE <capture nickname>
CONTROL TABLE SCHEMA <capture control schema>
```

创建存储过程搜索可以使用的文本索引

要使用存储过程搜索，需要在执行 CREATE INDEX 命令期间指定高速缓存选项。这通过将指定的所有数据移入主内存中来获取高性能。

但是，在用于搜索的第一次索引更新之前，确保表包含文档以避免更新未填充表的索引。这样可获得更好的建立索引性能和准确地估计高速缓存需求。

存储过程搜索可以快速返回与文档相关联的预定义数据。在 CREATE INDEX 命令中使用高速缓存表选项来进行定义。然后，ACTIVATE CACHE 命令将指定的数据移入内存高速缓存中。

注意

如果没有对视图创建文本索引，则 SQL 标量搜索函数也可以使用此文本索引。

在分布式 DB2 环境中，必须为存储过程在单个节点上为管理表显式指定表空间，并在此节点上显式调用。

为了确保连接至正确的节点，使用 DB2NODE 环境变量。

当为存储过程搜索创建文本索引时，必须确定和计算下列参数：

- 高速缓存的类型。
- 如何更新索引。
- Net Search Extender 可以使用的最大内存量 MAXIMUM CACHE SIZE。
- 后续文档更新需要的可用内存量 PCTFREE。注意，这只适用于增量更新。

提供了下列类型的高速缓存：

临时高速缓存

这是使用每个 DB2TEXT ACTIVATE CACHE 命令重新构建的，并且需要将数据从 DB2 表装入内存中。与激活持久高速缓存相比，这将花更长时间，尤其对大型索引更是如此。但是，它可能会稍微提高搜索性能。

持久高速缓存

这是保存在磁盘上的，可以在每个 DB2TEXT ACTIVATE CACHE 命令上通过操作系统快速映射至内存。在增量索引更新方案中，它必须保持激活状态，以允许索引与高速缓存之间同步。否则，下一个 DB2TEXT ACTIVATE CACHE 命令将从头开始重新创建高速缓存。

提供了下列方法来更新文本索引：

“更新时不重新创建索引”选项

避免删除表中的文档然后再重新插入这些文档，这是因为在高速缓存中不能重新使用已删除文档的槽。因此，应避免对已激活的索引更改键列。

这也称为增量更新。

“更新时重新创建索引”选项

每次更新时，此选项都将重新创建索引。在高速缓存列表达式中的任何可能位置使用 variable 数据类型。这将节省高速缓存空间。在 CACHE TABLE 子句中使用相应的转型表达式。

创建存储过程搜索可以使用的已高速缓存的文本索引

如果期望在初始索引激活之后插入超过 50% 的文档，则使用此选项。

Net Search Extender 提供了两个 SQL 函数来帮助您确定 CREATE INDEX 内存参数。这两个 SQL 函数为：MAXIMUM CACHE SIZE 和 PCTFREE。

对于增量更新和重新创建更新

以下命令将返回建议的 MAXIMUM CACHE SIZE 值（以兆字节（MB）计）：

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

以下命令将返回表的平均行长度参数：

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

注意，如果将更多文档插入到表中，则平均值可能会极大地更改。高速缓存列数目与在 DB2TEXT CREATE INDEX 命令的 CACHE TABLE 子句中使用的列表表达式数目相关。

有关附加信息，参见第 207 页的附录 B，『使用大量内存』。

仅对于增量更新

以下命令将根据实际文档数和最大文档数来返回建议的 PCTFREE 值。

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

实际文档数是第一次执行 ACTIVATE CACHE 命令（该命令将创建内存高速缓存）时表中的行数。

最大文档数是在运行下一个 DB2TEXT ACTIVATE 命令（对于临时高速缓存）或 DB2TEXT ACTIVATE CACHE RECREATE 命令（对于持久高速缓存）之前估计的表中的最大文档数。

如果每次更新时都重新创建索引，则将 PCTFREE 值设置为 0。

示例

假定您的表中具有 10000 行，而您不期望超过 20000 行。使用以下调用来计算您需要的 PCTFREE 值：

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

假定最大行大小为 20000，且在高速缓存中具有平均大小为 76 的 2 行。使用以下调用来返回该大小：

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

在确定合适的参数之后，可以使用以下调用来创建索引和高速缓存表：

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

在此示例中，除了通过将主内存用于快速返回结果表构建索引之外还构建了 docid 列。保留 10% 的高速缓存供以后的文档使用，并限制高速缓存最多为 5 MB。

更新文本索引

要对此索引进行搜索，需要更新该索引然后激活它。这会将指定的表高速缓存表达式从数据库复制到内存中。

如果在执行 ACTIVATE 或 UPDATE 操作期间，超过了 MAXIMUM CACHE SIZE 或 MAXIMUM NUMBER OF DOCUMENTS，则建议执行下列操作：

增量更新的 MAXIMUM CACHE SIZE

通过使用以下顺序的 DB2EXT 命令来重新构建高速缓存：DEACTIVATE CACHE、ALTER INDEX MAXIMUM CACHE SIZE 和 ACTIVATE CACHE RECREATE。

MAXIMUM NUMBER OF DOCUMENTS

对增量更新使用 DEACTIVATE CACHE 和 ACTIVATE CACHE RECREATE 命令，或者重新创建索引。

如果期望频繁更新文档，则考虑在 CACHE TABLE 子句中将固定大小的数据类型用于高速缓存列表表达式。以下示例说明如何在多次更新操作期间使用相同的高速缓存存储器：

```
CACHE TABLE(cast(C1 as char(20)), cast(substr(C2,1,10) as char(10)))...
```

对文本索引激活和取消激活高速缓存

在激活高速缓存之前，执行任何暂挂的增量更新来避免较差的 PCTFREE 计算。

要激活文本索引，使用以下命令：

```
db2text ACTIVATE CACHE FOR INDEX db2ext.comment FOR TEXT
```

此命令将指定的高速缓存表数据从数据库中检索出来，并将此数据存储在内存中。所用的时间取决于表的大小。

注意

如果在激活了索引时调用 update index，这也会更新高速缓存表。由于已删除的文档也会占用高速缓存中的槽，因此应确保将 PCTFREE 设置为足够大的值。

创建存储过程搜索可以使用的已高速缓存的文本索引

每当停止系统时，就需要重新执行 ACTIVATE CACHE 调用。如果使用持久高速缓存，则新的激活将更快。

注意

如果在没有激活持久高速缓存时进行了更新，则在调用激活期间将删除持久高速缓存然后再重新创建它。

为了节省资源，还应该取消激活当前不需要的任何索引。

要检查剩下了多少内存，使用以下调用：

```
db2text control show cache status for database cte index db2ext.comment
```

这将显示是否已经激活了索引以及指定的高速缓存空间还剩下多少。

有关视图的文本索引

当使用存储过程时，能够对视图创建文本索引。但是，它的一个主要缺点是不能对视图创建触发器，因此，不能识别底层基本表中的任何更改。

因此，对于增量索引更新，用户必须清楚已经添加、更新或删除了哪些文档以便使文本索引与数据库同步。为此，必须将所有更改添加至日志表。以下样本中说明了此过程：

1. 要创建基本表，使用以下命令：

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

2. 要添加一些条目，使用下列命令：

```
db2 "insert into sample values(1,'Claus','works in room 301')"  
db2 "insert into sample values(2,'Manja','is in the same office  
      as Juergen')"  
db2 "insert into sample values(2,'Juergen','has the longest way to  
      Raiko')"  
db2 "insert into sample values(3,'Raiko','is sitting in the office  
      besides Claus ')"
```

3. 要创建视图，使用以下命令：

```
db2 "create view sampleview as select key, comment from sample"
```

4. 使用下列命令来创建、更新和激活文本索引：


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

注意

需要指定高速缓存表以便能够对视图创建文本索引。要创建正确的日志表，必须对视图的索引指定键列。

在分布式 DB2 环境中，必须在单个节点上为管理表显式指定表空间，并在此节点上显式调用。

为了确保连接至正确的节点，使用 DB2NODE 环境变量。

- 5. 要更新表，使用下列命令：
db2 "insert into sample values(4,'Bernhard','is working in the same floor
as Manja, but not as Claus')"
db2 "insert into sample values(5,'Gunter','shares the office with Raiko')"
- 6. 然后，更新日志表。要获取日志表的名称，使用以下命令：
db2 "select INDSHEMA,INDNAME,LOGVIEWSCHEMA,LOGVIEWNAME
from db2ext.textindexes"

以下是日志表的布局:

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

要将条目添加到日志表中，使用下列命令:

```
db2 "insert into sample values(0,CURRENT TIMESTAMP,4)"
db2 "insert into sample values(0,CURRENT TIMESTAMP,5)"
```

第一个值用来描述操作（0 = 插入，1 = 更新，2 = 删除）。第二个值应当始终是 CURRENT TIMESTAMP，而最后一个值是已经插入的键。

- 7. 使用以下命令再次更新该索引：
db2text "update index indexview for text"

现在，可以使用存储过程对新值进行搜索了。

性能注意事项

要在建立索引期间提高性能，应考虑下列问题：

- 使用 VARCHAR 数据类型而不是 LONG VARCHAR 或 CLOB 来存储文本文档。
- 使用不同的硬盘来存储文本索引和数据库文件。
- 使用较小的主键列，例如，时间戳记和整数而不是 varchar 类型。
- 确保系统具有足够的实际内存可用于所有此类数据。如果没有足够的内存，则操作系统就会使用调页空间。这会降低搜索性能。

有关不同平台的配置内存需求的信息，参见第 207 页的附录 B，『使用大量内存』。

- 在自动或手工更新索引期间使用的更新落实计数参数在增量建立索引期间会降低建立索引的性能。注意，在初始更新过程中未使用该参数。

如果未设置落实计数，则不更新 db2ext.textindexes 中的 number_docs 参数。因此，要在更新过程中查看文档数，使用 CONTROL LIST 命令。有关信息，参见第 108 页的『CONTROL』。

注意

有关最新的性能提示，请访问以下 DB2 Net Search Extender Web 站点：
www.ibm.com/software/data/db2/extendors/netsearch/index.html

维护文本索引

本章描述如何维护文本索引和获取有关它们的有用信息。维护任务包括：

1. 更新和重组文本索引
2. 改变文本索引
3. 删除索引更新事件
4. 删除文本索引
5. 显示索引状态

可以在任何时候按任何顺序运行这些任务。本章还包括有关如何备份和复原索引和已启用的数据库的信息。

有关用来显示文本索引、目录名和更新的信息的命令，参见第 211 页的附录 C，『Net Search Extender 信息目录』。

还可以使用“DB2 控制中心”来维护文本索引。参见第 73 页的『维护文本索引』。

更新和重组文本索引

在第一次创建和更新文本索引之后，必须保持文本索引是最新的。例如，当将文本文档添加至数据库或者更改数据库中的现有文档时，必须对该文档建立索引以使索引内容与该数据库的内容同步。同样，当从数据库中删除文本文档时，必须从索引中除去其对应项。

如果未使用 `RECREATE INDEX ON UPDATE` 选项来创建文本索引，则触发器会自动将有关新的、已更改的或已删除的文档的信息存储在内部日志表中。因此，在下次进行索引更新时，将对日志表中引用的文档建立索引。有关视图的文本索引，参见第 48 页的『有关视图的文本索引』以获取附加信息。

如果在 `CREATE INDEX` 命令中指定了 `RECREATE` 选项，则将对每个更新完全重新构建索引。此选项不会创建任何日志表或触发器。

通常，按一定的时间间隔来更新索引。可以使用 `ALTER INDEX` 命令来更改现有索引的更新频率。

根据将要进行更新的时间和必须进行排队的最小文本更改数来指定索引更新频率。如果在给定的日期和时间内，日志表中没有足够的更改，则不会更新索引。

应当仔细地计划定期建立索引；对文本文档建立索引是一项既消耗时间又消耗资源的任务。所用的时间取决于很多因素。这些因素包括：文档大小、自从前一次索引更新以来已经添加或更改了的文本文档数以及处理器的能力。

注意

在 DB2 表上，在下列情况下可能会发生回滚和死锁情况：

- 高更新频率
- 高频率更改事务
- 长事务

`UPDATE INDEX` 命令允许您一旦请求就立即更新索引。

总结

何时删除文本索引

当必须立即更新索引而不等待进行定期建立索引时。

命令 UPDATE INDEX

权限 对表的 CONTROL 权限

以下命令将更新索引:

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

当已将几个文本文档添加至数据库并且想要立即搜索它们时，此命令很有用。

要确定是否需要进行手工重组，通过使用以下命令来查询 db2ext.textindexes 视图:

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

如果指定了 MANUAL REORGANIZATION 并且经常更新一列，则更新过程将变慢。要手工进行重组，使用以下命令:

```
dbtext UPDATE INDEX comment FOR TEXT reorganize
```

但是，如果在 CREATE INDEX 期间指定了 AUTOMATIC REORGANIZE，则在需要时将自动重组索引。

改变文本索引

总结

何时删除文本索引

当必须更改更新频率或者索引目录和工作目录时。

命令 ALTER INDEX

权限 对表的 CONTROL 权限

使用此命令来更改索引工作目录、索引的更新频率或者高速缓存特征（主要是 MAXIMUM CACHE SIZE 或 PCTFREE）。如果不指定更新频率，则当前设置将保持不变。如果正在运行索引更新或搜索，则会显示错误消息。这表明当前锁定了索引，不能执行任何更改。

以下示例将更改索引的更新频率。

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

在此示例中，如果队列中至少有 100 个文本文档，则将在星期一至星期五的 12:00 或 15:00 更新索引。

使用以下命令来停止索引的定期更新：

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

如果索引是从一个目录复制至另一个目录的，则在此过程中索引是锁定的。但是，在复制之后，索引会解锁，并且可以再次使用它。

清除索引事件

总结

何时删除文本索引

当您不再需要索引的事件表中的消息时。

命令 `CLEAR EVENTS FOR INDEX`

权限 对表的 `CONTROL` 权限

有关建立索引事件的信息（例如，更新的开始时间和结束时间、已建立索引的文档数或者更新期间发生的文档错误）都存储在索引的事件表中。这可以帮助您确定问题的原因。当不再需要这些消息时，可以删除它们。

以下示例将从指定的文本索引中删除消息：

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

删除文本索引

总结

何时删除文本索引

当您不再打算在文本列中进行文本搜索时。

命令 DROP INDEX FOR TEXT

权限 对表的 CONTROL 权限

示例:

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

当删除文本索引时，也删除了下列表和视图:

- 日志表和视图
- 事件表和视图
- 日志表触发器（如果存在的话）

注意

始终要在删除对表的索引之后才删除表。如果先删除表，则索引仍然存在。

查看文本索引状态

要获取有关数据库中的当前文本索引的信息，请使用视图。例如，如果想知道当前的数据库缺省值，使用以下命令:

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

要获取当前可用的索引、它们相应的表和已建立索引的文档数的信息，请使用以下命令:

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

使用以下命令来获取有关特定索引的格式的信息:

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

有关进一步的信息，参见第 211 页的附录 C，『Net Search Extender 信息目录』。

如果未设置落实计数，则不更新 `db2ext.textindexes` 中的 `number_docs` 参数。要在更新过程中查看文档数，使用以下命令：

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

有关进一步的信息，参见第 108 页的『CONTROL』。

备份与复原索引

执行下列步骤来备份已启用的数据库和由 DB2 Net Search Extender 创建的文本索引：

1. 要了解 DB2 Net Search Extender 已经创建了哪些索引以及将这些索引存储在何处，可对 `db2ext.textindexes` 视图调用 Select 语句：

```
db2 "select indschema, indname, indexdirectory from db2ext.textindexes"
```

2. 确保没有任何索引更新在运行，然后使用以下命令来停止 DB2 Net Search Extender 服务：

```
db2text stop
```

3. 在备份数据库之后，备份索引目录和子目录。
4. 使用以下命令重新启动 DB2 Net Search Extender 服务：

```
db2text start
```

执行下列步骤来复原已启用的数据库和由 DB2 Net Search Extender 创建的文本索引：

1. 使用以下命令来停止 DB2 Net Search Extender：
- ```
db2text stop
```
2. 将索引目录的备份副本复原到以前的同一路径中。
  3. 使用以下命令重新启动 DB2 Net Search Extender：
- ```
db2text start
```


第 7 章 使用 DB2 控制中心

使用“DB2 控制中心”来管理 DB2 Net Search Extender 管理函数、DB2 实例、数据库和数据库对象（例如，表、视图和用户组）。

可以对不同的“DB2 控制中心”对象调用命令，例如：

- 实例对象
- 数据库对象
- 文本索引对象

“DB2 控制中心”的主要元素包括菜单栏、工具栏、对象树和内容窗格。

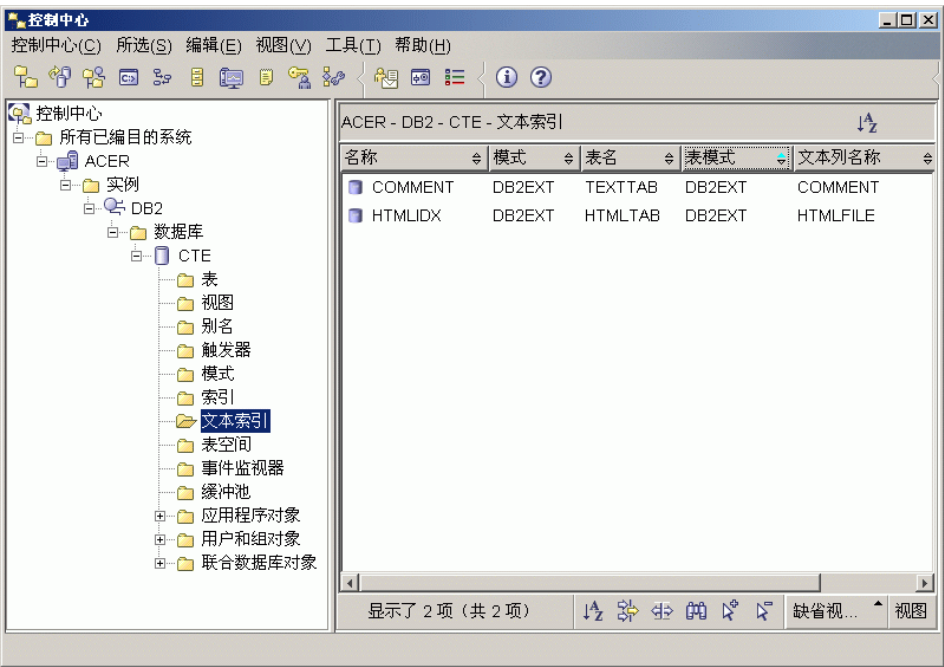


图 6. DB2 控制中心

也可以使用命令行。有关更多信息，参见下列各章：

- 第 35 页的第 6 章，『创建和维护文本索引』
- 第 29 页的第 5 章，『Net Search Extender 实例服务』

注意

要使用 Net Search Extender 的示例和“DB2 控制中心”，必须在数据库服务器上安装有效的 Net Search Extender 许可证。

在本章中只能找到 DB2 Net Search Extender 建立索引和管理函数。有关使用“DB2 控制中心”的信息，参见第 viii 页的『相关信息』。

启动和停止“DB2 Net Search Extender 实例服务”

从对象树中，单击系统以显示可用的实例。突出显示该实例并进行右键单击以显示实例对象弹出菜单。突出显示 **Net Search Extender**，并从弹出菜单中选择下列命令之一：

启动 DB2 Net Search Extender 实例服务

如果尚未启动实例服务，则会启动实例服务。

停止 DB2 Net Search Extender 实例服务

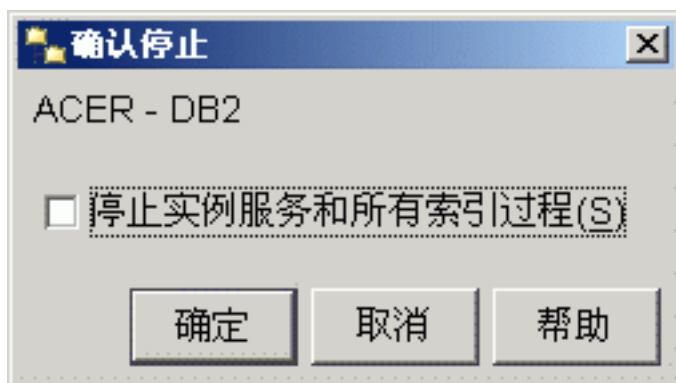


图 7. “停止 Net Search Extender 服务”对话框

这会显示一个对话框。使用复选框来停止实例服务和索引进程。在命令语法中，这称为 **FORCE** 选项。单击**确定**按钮。

实例状态

这会显示一个对话框，该对话框显示实例的状态。

有关进一步的信息，参见第 107 页的第 11 章，『实例所有者的管理命令』。

启用数据库

在对象树中，单击实例对象以显示可用的数据库。突出显示该数据库并进行右键单击以显示弹出菜单。突出显示 **Net Search Extender**，并从展开的菜单中选择下列命令之一：

启用数据库的文本搜索

如果未启用数据库的文本搜索，则会显示一个对话框。单击**确定**按钮以启用数据库。如果已启用数据库，则会显示一个消息框。

禁用数据库的文本搜索

如果未禁用数据库的文本搜索，则会显示一个对话框。单击**确定**按钮以禁用数据库。如果已禁用数据库，则会显示一个消息框。

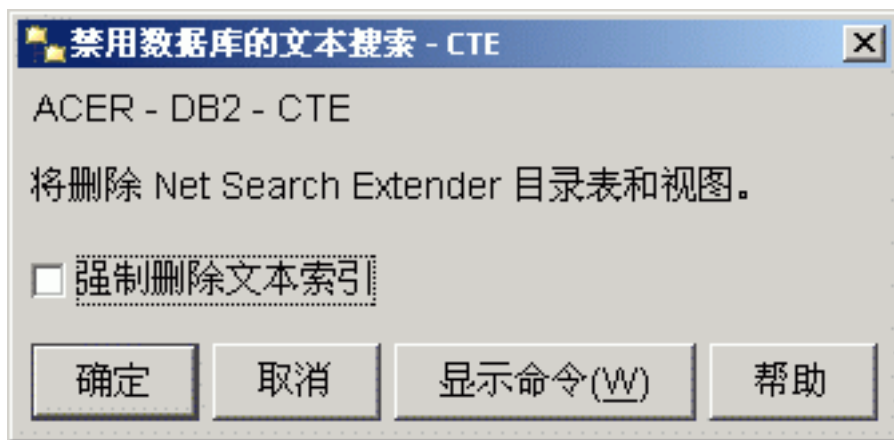


图 8. “禁用数据库的文本搜索”对话框

如果想要禁用数据库并删除所有文本索引，则单击该复选框。

更改 Data Link 返回大小

这会显示一个对话框，该对话框显示当前的 Data Link 返回大小。以千字节（KB）为单位输入新的 Data Link 值，并单击**确定**按钮。

有关进一步的信息，参见第 113 页的第 12 章，『数据库管理员的管理命令』。

注意，在所有对话框中，**显示命令**按钮将显示命令行备用项。

文本索引管理

在对象树中，可以在数据库对象下面查看文本索引对象。单击文本索引对象以在内容窗格中查看文本索引。

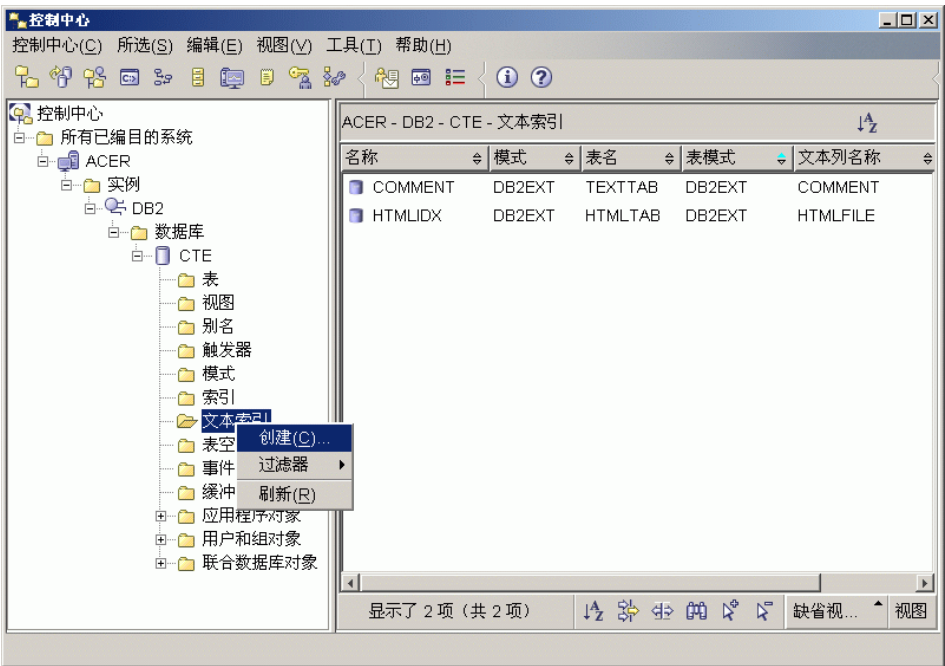


图 9. DB2 控制中心

右键单击文本索引对象，并从弹出菜单中选择下列命令之一：

- 创建** 这会显示用于创建文本索引的向导。有关更多信息，参见第 61 页的『创建文本索引』。
- 过滤** 这会显示一个对话框，可在该对话框中选择将哪些文本索引对象显示在控制窗格视图中。
- 刷新** 这会刷新对象树和控制窗格中的信息。

要维护文本索引，参见第 73 页的『维护文本索引』。

注意

要想不通过使用右键单击选项来访问实例、数据库和文本索引对象命令，则单击**所选菜单命令**，并突出显示 **Net Search Extender** 以访问相关命令。

在创建文本索引之前，确保已经考虑了第 25 页的第 4 章，『规划』中的先决条件。

其它建立索引先决条件包括：

- 启动 DB2 Net Search Extender 实例服务
- 启用数据库

创建文本索引

选择**创建**命令，将弹出“创建文本索引向导”。使用该向导来为许多面板中的文本索引指定配置选项。要在各面板之间移动，输入所有必填信息，并单击**下一步**按钮，直到**完成**按钮可用为止。单击**完成**按钮以创建文本索引。

要对视图创建文本索引，使用第 130 页的『CREATE INDEX』中描述的 CREATE INDEX 命令。

“名称”面板

此面板允许您指定文本索引的模式和名称。还可以指定文本索引文件的工作目录和索引目录。为管理表空间上的索引创建管理表。

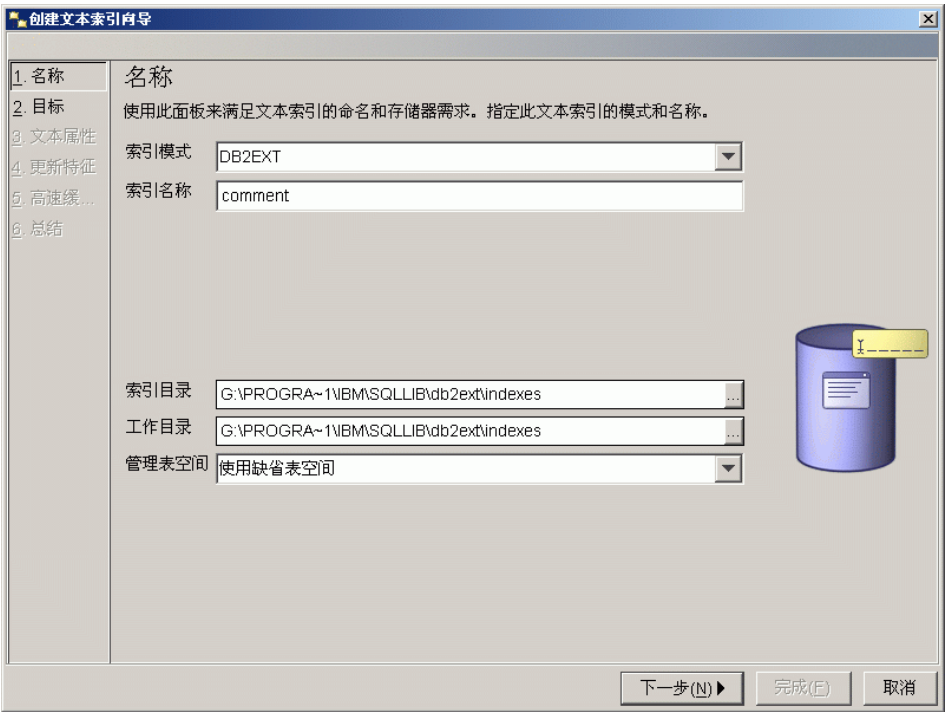


图 10. 创建文本索引向导：“名称”面板

以下是对该面板中各字段的描述：

表 1. “名称”面板文本字段

字段名	必填 / 可选	缺省值	描述
索引模式	必填	用户标识	选择文本索引的模式名。这是特定于索引的管理表的 DB2 模式名。
索引名称	必填	不适用	为文本索引输入有效的 DB2 索引名。借助索引模式，索引名将唯一地标识数据库中的全文本索引。
索引目录	可选	参见路径名	指定将用来存储文本索引的目录路径。对于 DB2 所有者用户标识，目录必须存在，并且具有读、写和运行许可权。
工作目录	可选	参见路径名	指定在执行搜索和管理操作期间将用来存储临时文件的工作目录。对于 DB2 所有者用户标识，目录必须存在，并且具有读、写和运行许可权。

表 1. “名称” 面板文本字段 (续)

管理表空间	可选	使用缺省表空间	选择文本索引管理表的表空间名。必须在用户表的表空间所在的同一个节点组上定义管理表空间。
-------	----	---------	---

“目标” 面板

此面板允许您指定表或昵称表的模式和名称以及包含想要建立索引的数据的文本列的名称。可以使用变换函数来修改文本列的内容。除了文本列之外，如果想要对表列表表达式的内容建立索引，则还可以指定数字属性。

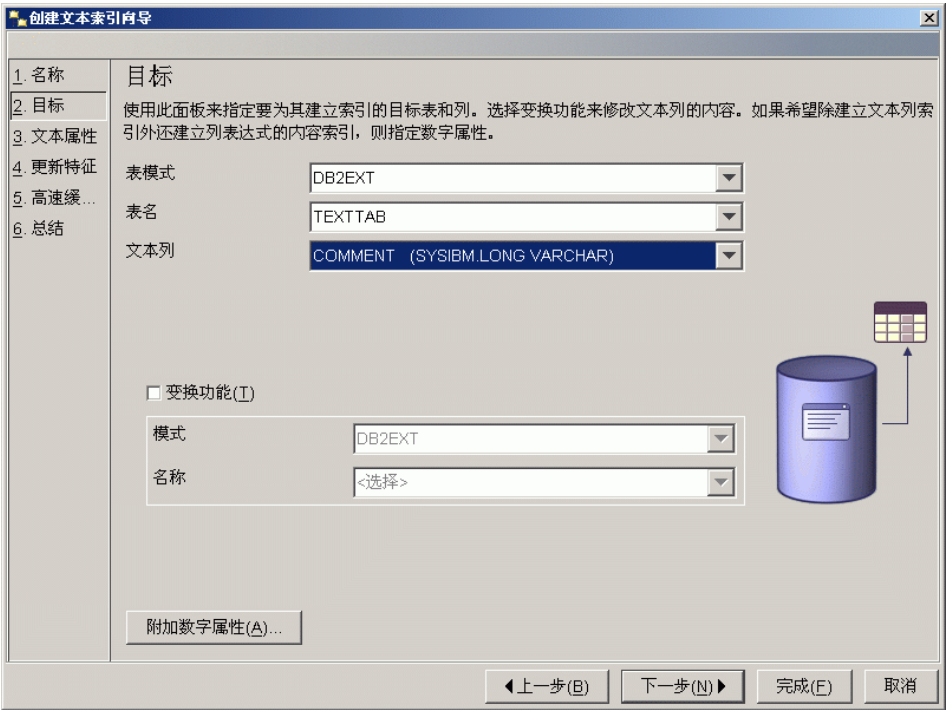


图 11. 创建文本索引向导: “目标” 面板

以下是对该面板中各字段的描述:

表 2. “目标” 面板文本字段

字段名	必填 / 可选	缺省值	描述
表模式 (1)	必填	用户标识	选择正在对其创建文本索引的表或昵称表的模式。

表 2. “目标” 面板文本字段 (续)

表名 (2)	必填	不适用	选择正在对其创建索引的表或昵称表的名称。表必须具有主键。
文本列 (3)	必填	不适用	选择用于创建文本索引的列的名称。必须将列变换为下列类型之一，或者它们必须属于下列类型之一：CHAR（对于位数据）、VARCHAR（对于位数据）、LONG VARCHAR（对于位数据）、CLOB、DBCLOB、BLOB、GRAPHIC、VARGRAPHIC、LONG VARGRAPHIC 和 DATALINK。
变换函数	可选	禁用	选择此项以使用变换函数。
变换函数：模式	必填（如果选择了函数的话）	用户标识	选择用来存取文本文档的 UDF 的模式。
变换函数：名称	同上	不适用	选择用来存取文本文档的 UDF 的名称。

注意，只能按次序指定表模式（1）、表名（2）和文本列（3）。

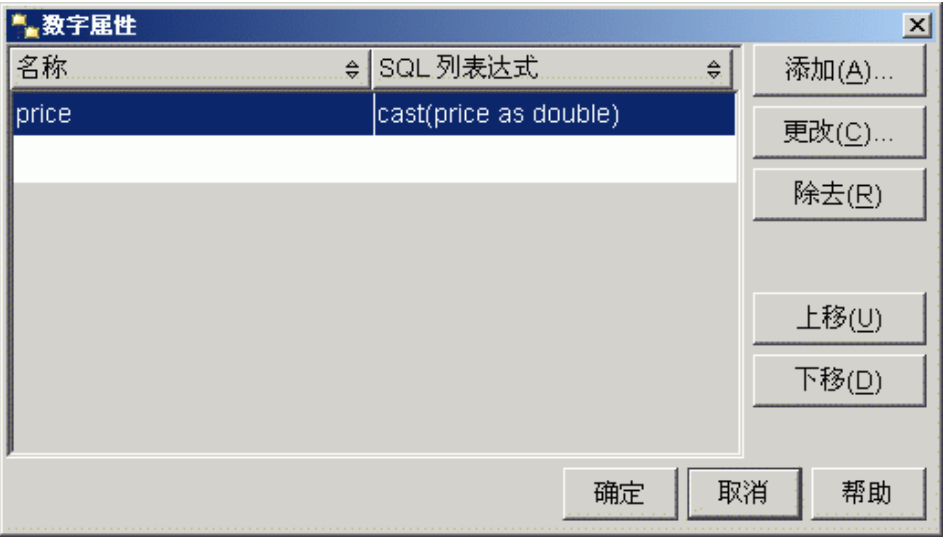


图 12. “数字属性” 对话框

要查看或添加属性，单击**数字属性**按钮。将显示一个窗口。要对索引添加数字属性，单击**添加**按钮，将显示另一个窗口。指定属性的 SQL 列表表达式和名称。

或者，选择一种属性，并按适当的按钮来更改、移动或删除条目。

说明数字属性

使用“数字属性”来除了对文本列之外还对列表表达式建立索引。例如，如果想要除了对文本列之外还对类型为 `TIMESTAMP` 的列日期建立索引，则应指定数字属性 `"cast(julian_day(date) as double)"`，并为该属性指定名称。

如果想要在搜索查询中使用数字表达式，则应指定数字属性。如果您正在使用 `SQL` 查询进行搜索，则可以使用组合搜索而不使用数字属性，例如：`WHERE numattrib = 123 AND contains('...')`。如果正在使用存储过程进行搜索，则必须为文本文档指定数字属性（如果想要搜索文本文档的附加信息的话）。

“文本属性”面板

此面板允许您指定文本文档的语言和格式。如果文档与数据库不是采用相同的 `CCSID`，并且文本列属于二进制类型，则应指定 `CCSID`。注意，最初已经选择了数据库 `CCSID`。如果文档具有 `GPP`、`HTML`、`Outside-In` 或 `XML` 结构化格式，则可以指定文档模型。

注意

在格式列表框中，`Outside-In` 过滤格式称为 `INSO`。

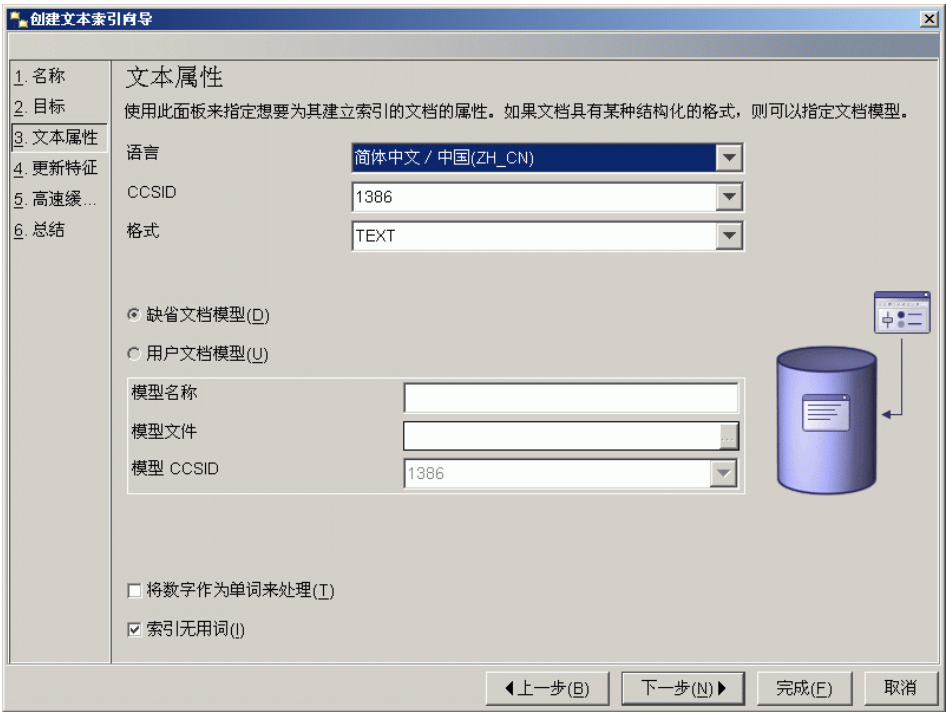


图 13. 创建文本索引向导：“文本属性”面板

以下是对该面板中各字段的描述:

表 3. “文本属性”面板文本字段

字段名	必填 / 可选	缺省值	描述
语言	可选	EN_US	当对文档建立索引时，选择一种语言以确定句子结束定界符和段落结束定界符。
CCSID	可选	数据库的 CCSID	选择用于对文本文档建立索引的 CCSID。
格式	可选	TEXT	选择文本文档格式： HTML、XML、TEXT、INSO 或 GPP。
缺省文档模型	可选	启用	使用缺省文档模型。
用户文档模型	可选	禁用	使用您的文档模型。
模型名称	必填（如果选择了“用户文档模型”的话）	不适用	输入文档模型的名称。对于 HTML、XML、Outside-In 和 GPP 格式，可以指定文档模型。注意，只有在模型文件中才能找到该名称。

表 3. “文本属性” 面板文本字段 (续)

模型文件	同上	不适用	指定文档模型文件。DB2 实例所有者必须可以读取该文件。
模型 CCSID	同上	数据库 CCSID	选择 CCSID 以解释文档模型文件的内容。
将数字当作词语	可选	禁用	选择此项以将一系列数字解释为独立的词语，即使它们与字符相邻也是这样。
索引无用词	可选	启用	选择此项以启用特定于语言的无用词处理。 sql1lib/db2ext/resources 目录中的 <language>.tsw 包含无用词列表。

“更新特征” 面板

此面板允许您指定索引是以增量方式更新还是从头开始重新创建。可以指定更新设置，以便索引会在指定时间自动更新。

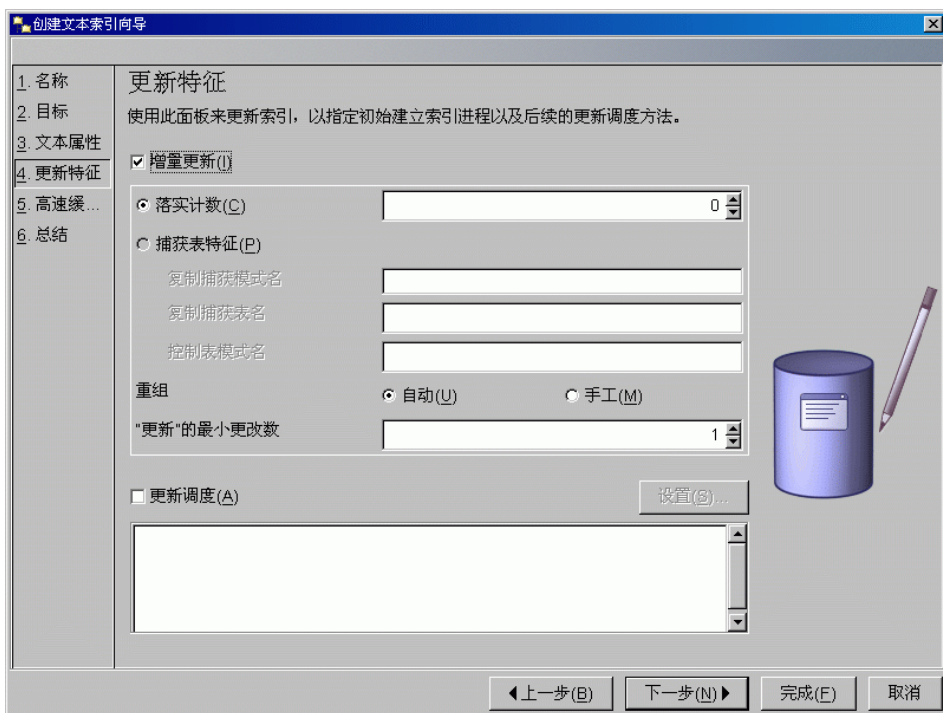


图 14. 创建文本索引向导: “更新特征” 面板

以下是对该面板中各字段的描述:

表 4. “更新特征” 面板文本字段

字段名	必填 / 可选	缺省值	描述
增量更新	可选	启用	选择此项以进行增量索引更新。如果不启用该复选框，则在执行更新操作时将重新创建索引。
落实计数	可选	0	在一个事务中执行更新期间所处理的更改数。 落实计数对性能有影响。有关信息，参见第 50 页的『性能注意事项』。
捕获表特征	可选	不适用	选择此项以使用复制捕获表来捕获对源表的更改。复制捕获表必须是“捕获数据”（CD）表或“捕获更改数据”（CCD）表，并替换 DB2 Net Search Extender 生成的日志表。
复制捕获模式名	可选	用户标识	复制捕获表的模式名。注意，先前必须已经使用“DB2 复制”创建了该表。
复制捕获表名	必填（如果启用了捕获表特征的话）	不适用	复制捕获表的表名。注意，先前必须已经使用“DB2 复制”创建了该表。
控制表模式名	必填（如果启用了捕获表特征的话）	不适用	控制表模式名。注意，先前必须已经使用“DB2 复制”创建了控制表。
自动或手工重组单选按钮	可选或必填	启用 / 禁用	自动或手工完成索引重组。
“更新”的最小更改数	可选	1	指定在指定时间对索引进行增量更新之前对文本文档所做的更改的最小数目。
更新调度	可选	禁用	选择此项以添加自动更新设置。

要添加索引更新设置，单击**设置**按钮。注意，仅当您选择了**更新调度表**时才会启用此按钮。在此对话框中，选择更新时间的天、小时和分钟。注意，如果选择了很多天，则会在所选择的每一天的同一时间进行更新。

“高速缓存表” 面板

此面板除了允许您指定索引之外，还允许指定高速缓存的表。可以指定要高速缓存的结果列，并且可以使用存储过程来搜索高速缓存。还可以指定其它高速缓存

参数，例如，类型、最大大小以及在初始建立索引期间检索用户表的内容所采用的次序。

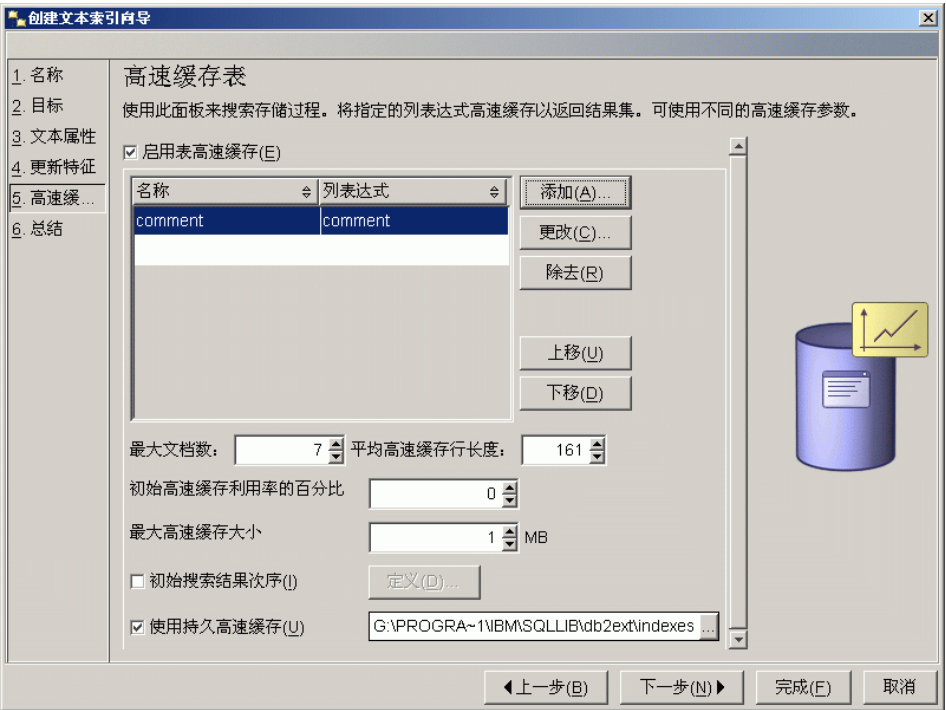


图 15. 创建文本索引向导：“高速缓存表”面板

以下是对该面板中各字段的描述：

表 5. “结果高速缓存” 面板文本字段

字段名	必填 / 可选	缺省值	描述
启用表高速缓存	可选	禁用	选择此项以启用高速缓存表的构建。
结果列表	必填（如果选择了“启用表高速缓存”的话）	不适用	显示用来指定搜索结果列的 SQL 列表表达式的列表。
最大文档数	必填	表的行计数	参见下面这一节：确定高速缓存利用率 and 高速缓存大小。
平均高速缓存行长度	必填	不适用	参见下面这一节：确定高速缓存利用率 and 高速缓存大小。

表 5. “结果高速缓存” 面板文本字段 (续)

初始高速缓存利用率的百分比	可选	50%	选择保留的用于附加文档的高速缓存的百分比。
最大高速缓存大小	可选	不适用	指定在索引激活期间构建的高速缓存表的最大大小。如果该数目太小，则将无法激活。
初始搜索结果次序	可选	禁用	选择此项以定义搜索结果次序。通过跳过对全文本搜索结果进行动态排序，文档将按照在高速缓存表中相同的建立索引次序返回。进行增量更新之后就不能确保该此次序了。
使用持久高速缓存	可选	启用	在取消激活或系统重新引导之后，此选项将启用执行快速激活。注意，必须指定持久高速缓存的目录路径。如果高速缓存是临时的，则保持禁用状态。

确定高速缓存利用率和高速缓存大小

初始高速缓存利用率的百分比指定要保留给附加文档使用的高速缓存的百分比。最大高速缓存大小指定在激活高速缓存期间要构建的高速缓存表的最大大小。这些选项取决于下列因素：

- 表中的实际文档数。
- 期望的更新数。
- 想要高速缓存的 SQL 表达式的平均大小。

可以输入初始高速缓存利用率的百分比和最大高速缓存大小的建议值。或者，也可以每当在最大文档数或平均高速缓存行长度字段中输入值时就计算这些值。

最大文档数值初始设置为表的行计数。根据文档数和期望的更改数来修改此值。包括所有文档更新、添加和删除。

当将 SQL 表达式添加至高速缓存表列表时，就会根据结果的长度来计算平均高速缓存行长度。由于这是基于表中的行数的，所以计算可能要花相当长的时间。如果您知道此值通常比较小，则修改该值。

例如，如果一个表具有 10 个条目，列表式的总计为 100，则这些值是初始设置的。如果期望最大文档数（包括已删除的文档）为 10000，则输入此数字。如果您知道列表式平均起来小于计算的值（例如，VARCHAR(100) 和已填写的文本大小 10），则使用此数字作为平均行大小。

要定义初始搜索结果次序，单击**定义**按钮。注意，仅当选择了“初始搜索结果次序”复选框时才会启用此按钮。一个对话框将显示指定的所有 SQL 列表表达式。要添加结果次序，单击**添加**按钮，并在对话框中，指定 SQL 结果次序。

要更改、移动或删除条目，选择表达式并单击适当的按钮。

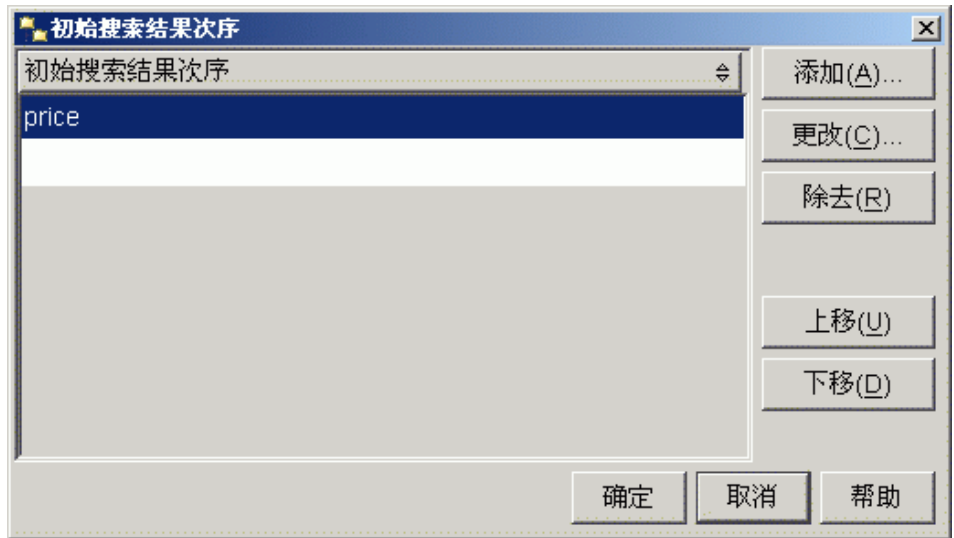


图 16. “初始搜索结果次序”对话框

要添加 SQL 列表表达式，单击“结果列”表旁边的**添加**按钮。在该对话框中，指定结果列表表达式和名称。

要更改或删除条目，单击将启用适当按钮的列表表达式。



图 17. “更改列表表达式” 对话框

“总结” 面板

此面板对先前选择的参数进行了概述。

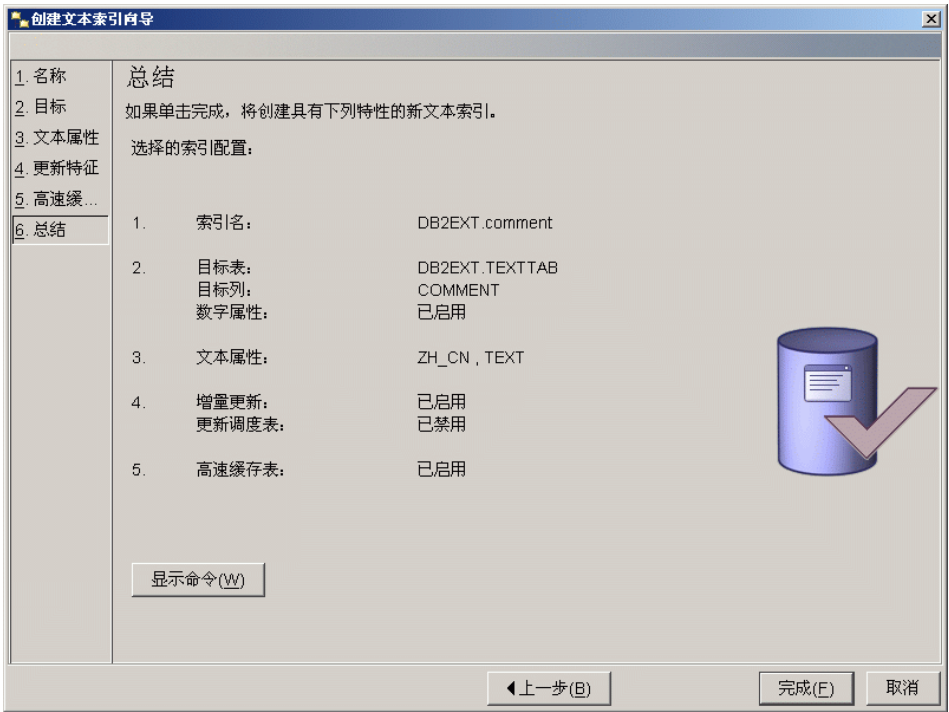


图 18. 创建文本索引向导：“总结”面板

单击**显示命令**按钮以查看当单击**完成**按钮时将运行的命令。此操作将创建文本索引。

维护文本索引

要维护文本索引，在控制窗格中选择文本索引，并单击**选择**菜单命令。可以从菜单中选择下列命令之一：

1. ALTER 命令，用于改变文本索引。
2. DROP 命令，用于删除文本索引。
3. UPDATE 命令，用于更新文本索引。
4. SHOW INDEX EVENTS 命令，用于显示索引事件。
5. ACTIVATE INDEX MEMORY 命令，用于激活索引高速缓存。
6. DEACTIVATE INDEX MEMORY 命令，用于取消激活索引高速缓存。
7. SHOW STATUS 命令，用于显示索引状态。

注意，仅当使用高速缓存选项创建索引时，才会显示“激活”和“取消激活”命令。

改变文本索引

选择**改变**命令，一个对话框就会显示一系列面板。这些面板对文本索引参数进行了概述。注意，**不能**更改所有参数。



图 19. “改变索引”对话框：“名称”选项卡

名称面板显示索引的名称和存储器配置。可以更改索引目录和工作目录。

目标面板显示索引的目标和数字属性设置。不能更改这些配置。

文本面板显示文本文档配置。不能更改这些设置。



图 20. “改变索引”对话框：“更新”选项卡

更新面板显示索引配置的更新特征。可以改变更新调度。如果索引是为增量更新创建的，则还可以修改最小更改数。如果索引是使用落实计数选项创建的，则还可以修改落实计数值。

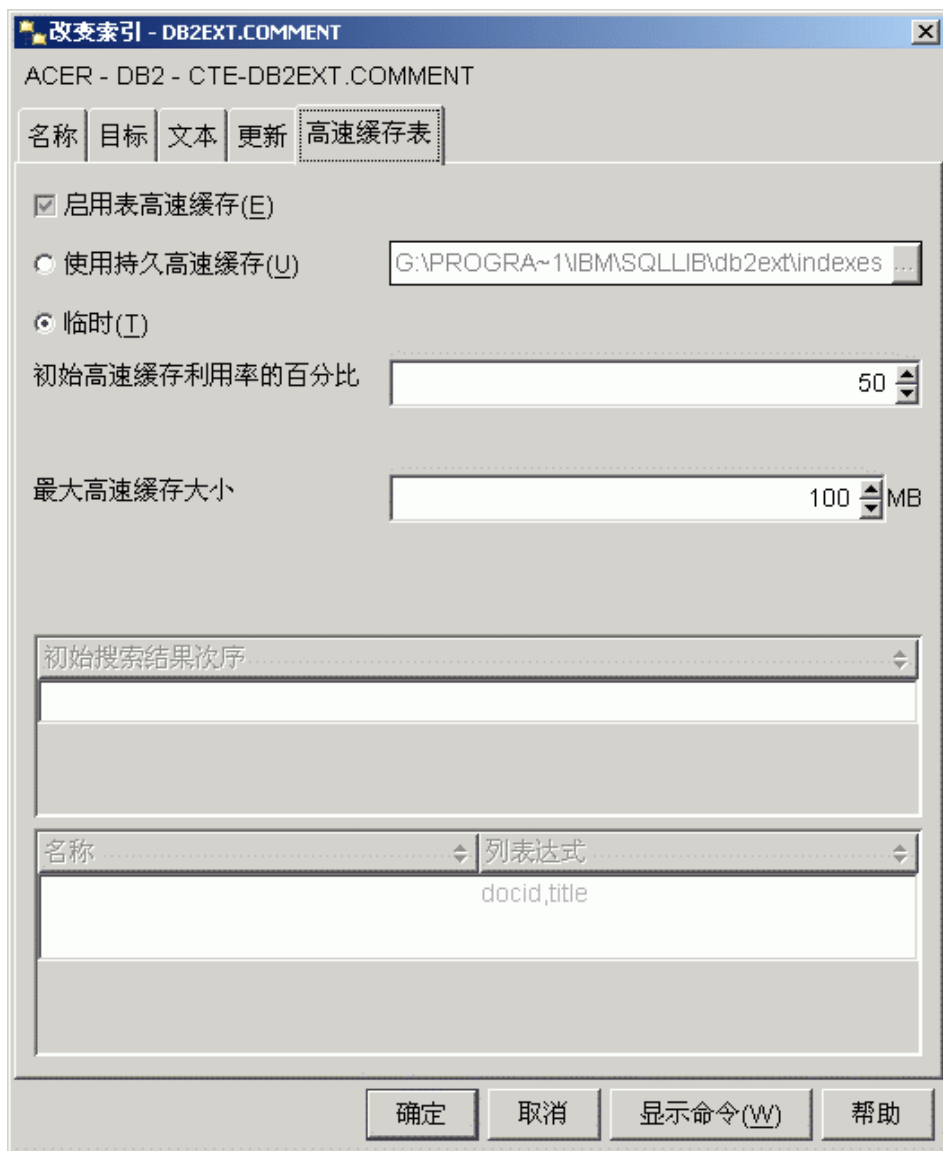


图 21. “改变索引”对话框：“高速缓存表”选项卡

高速缓存表面板显示了高速缓存选项设置。如果已经启用了结果高速缓存，则可以修改持久目录，也可以使索引高速缓存是临时的。还可以更改“最大高速缓存大小”和“初始高速缓存利用率的百分比”。

删除文本索引

选择删除命令，一个对话框就会显示可用的文本索引。

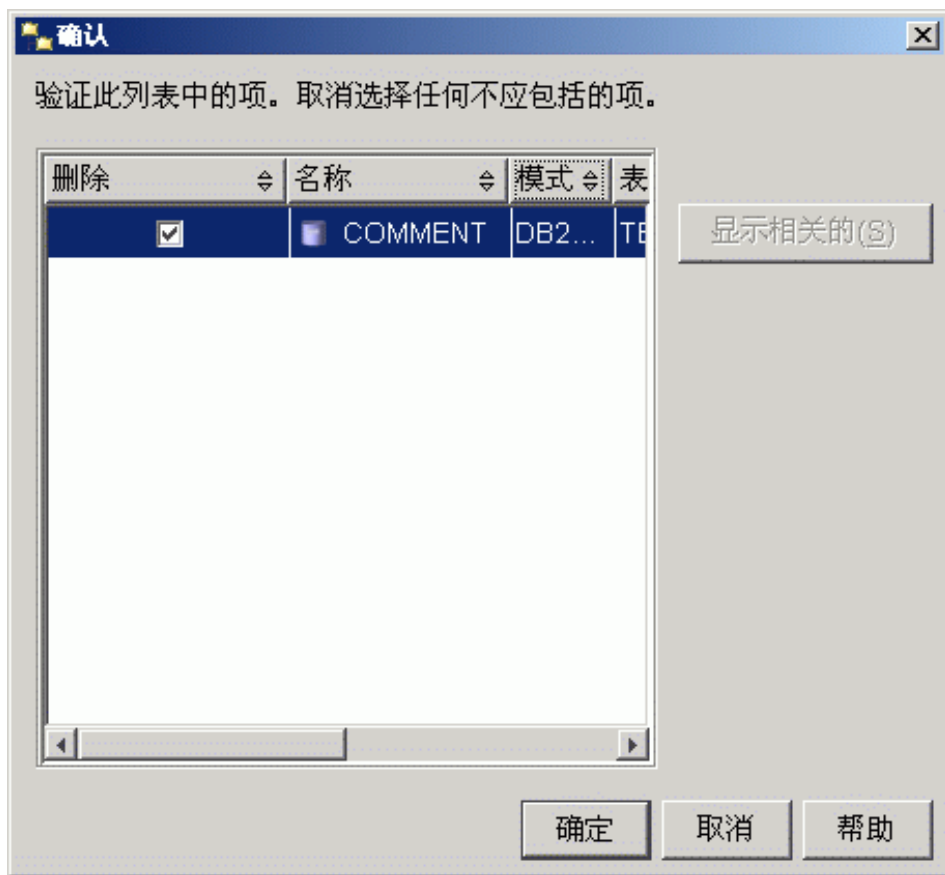


图 22. “删除索引”对话框

选择索引，并单击**确定**按钮。

更新文本索引

选择更新命令，一个对话框就会显示许多更新选项。



图 23. “更新索引”对话框

可以指定更新操作的落实计数。如果想要更新在创建或改变索引期间指定的最小值，则选择“最小的更改数”复选框。要重组索引，选择该复选框。注意，如果建议进行重组，则会启用该复选框。

显示索引事件

选择显示索引事件命令，事件表的内容就会显示在对话框中。

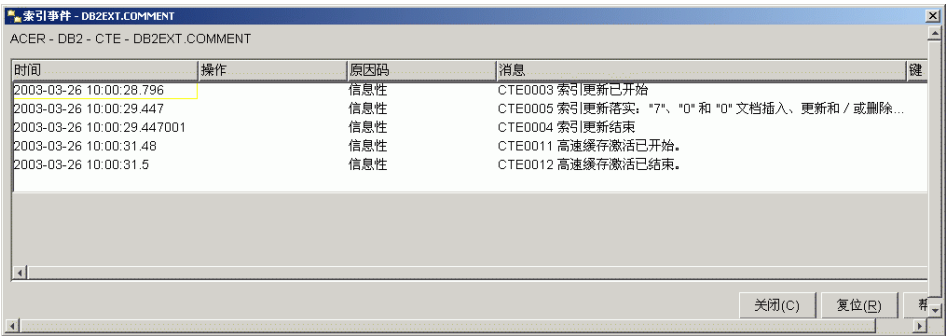


图 24. “索引事件” 对话框

要清除索引事件，单击**复位**按钮。

激活文本索引高速缓存

选择**激活索引内存**命令，将显示一个对话框。

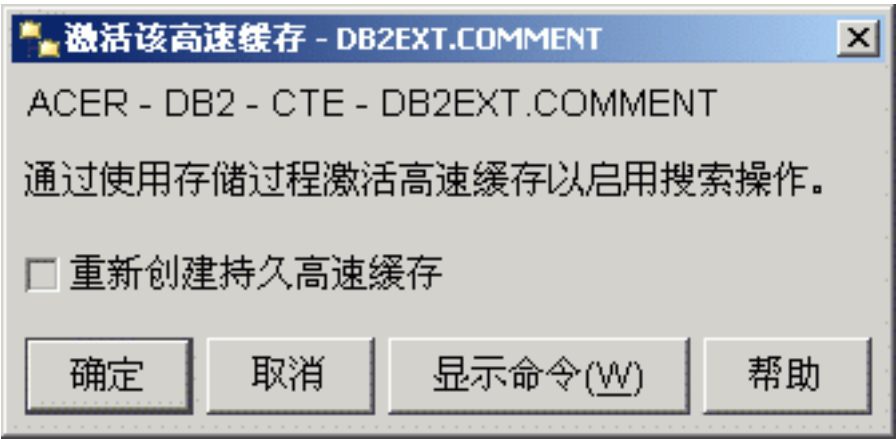


图 25. “激活高速缓存” 对话框

要激活高速缓存，单击**确定**按钮。如果想要从头开始构建高速缓存，则选择该复选框。

取消激活文本索引高速缓存

选择**取消激活索引内存**命令，将显示一个对话框。

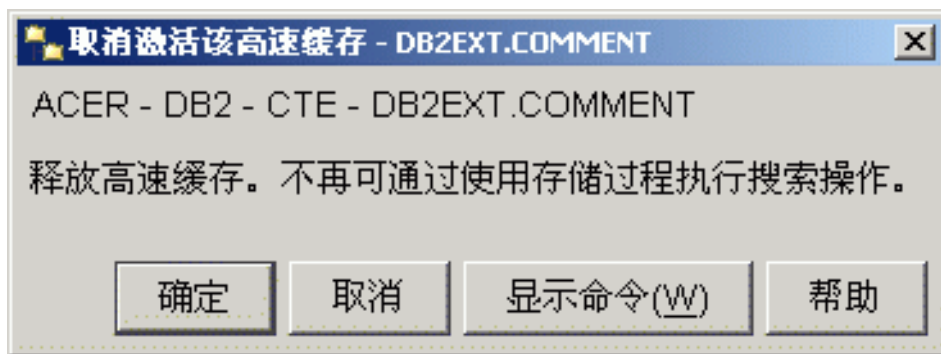


图 26. “取消激活高速缓存表”对话框

要释放高速缓存，单击**确定**按钮。

显示索引状态

选择**显示状态**命令，一个对话框将显示文本索引的状态。

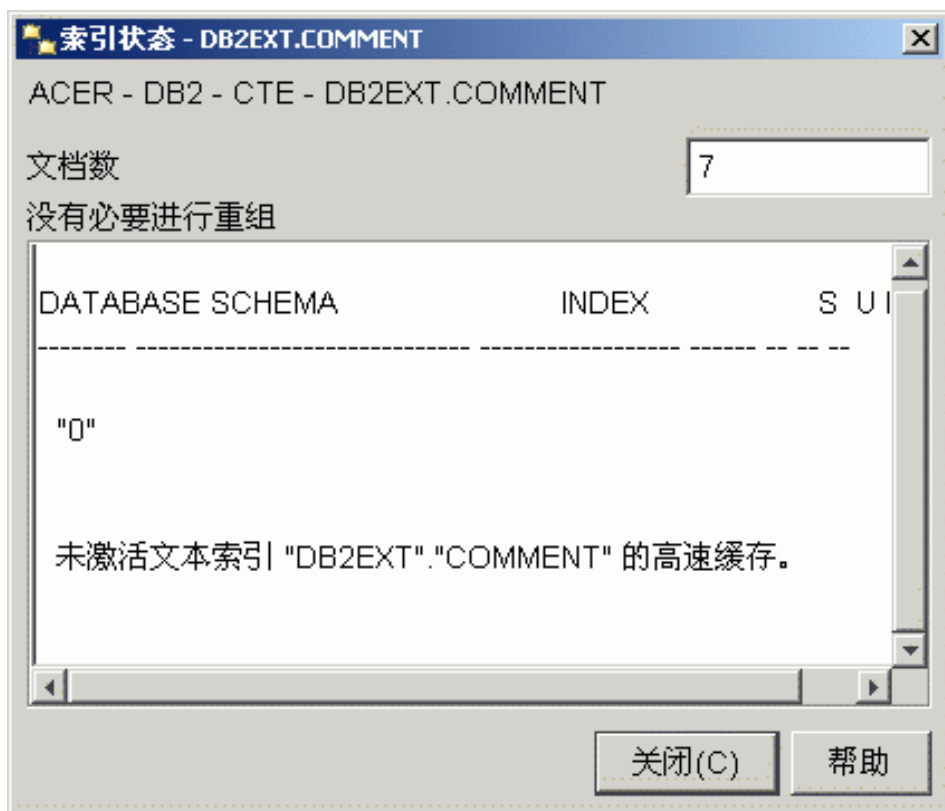


图 27. “索引状态”对话框

这包括有关已建立索引的文档数、建议的重组标志和附加索引信息的信息。

第 8 章 搜索

DB2 Net Search Extender 提供了下列方法来搜索文本：

SQL 标量搜索函数

它们使您能够在 SQL 查询中包括文本搜索子查询。除了 SQL 中通常提供的那些函数之外，Net Search Extender 还提供了这些函数。

存储过程搜索函数

这使您能够返回预定义的高速缓存结果表。

SQL 表值函数

可以使用与存储过程相似的方法来使用此搜索。

对于 SQL 标量搜索函数，本章描述了下列方面：

- 使用 CONTAINS、NUMBEROFMATCHES 和 SCORE 函数来搜索文本。
有关语法的描述，参考第 167 页的第 15 章，『SQL 标量搜索函数和 SQL 表值函数』。
- 通过将示例与 CONTAINS 函数配合使用来指定搜索自变量。
有关语法的描述，参考第 159 页的第 14 章，『搜索自变量的语法』。

对于存储过程搜索函数，本章描述了下列方面：

- 使用存储过程搜索来搜索文本。
- 要指定搜索自变量，参考第 159 页的第 14 章，『搜索自变量的语法』以获取对参数的描述。

对于“SQL 表值函数”，本章描述了下列方面：

- 使用“SQL 表值函数”和 HIGHLIGHT 函数来搜索文本。
有关语法的描述，参考第 167 页的第 15 章，『SQL 标量搜索函数和 SQL 表值函数』。
- 要指定搜索自变量，参考第 159 页的第 14 章，『搜索自变量的语法』以获取对参数的描述。

还有一些您可能需要考虑的搜索性能注意事项的信息。

在进行搜索之前，务必考虑第 35 页的第 6 章，『创建和维护文本索引』中描述的涉及到不同数据类型的所有适当的建立索引步骤。

注意

系统外壳程序将解释诸如 ?、(、)、*、! 和 " 之类的特殊字符。因此，如果命令中包含这些字符，则必须使用引号或转义字符。

以下是使用特殊字符的 UNIX 命令的一个示例：

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

使用 SQL 标量搜索函数搜索文本

通过使用一些示例，本节描述了如何按下列方式使用 SQL 标量搜索函数：

- 使用 CONTAINS 函数进行查询。
- 使用函数 NUMBEROFMATCHES 确定在文本文档中找到多少个匹配项。
- 使用 SCORE 函数来获取找到的文本文档的关联性。

有关语法的描述，参考第 167 页的第 15 章，『SQL 标量搜索函数和 SQL 表值函数』。

进行查询

此示例演示 CONTAINS 函数如何搜索 texttab 表中的 comment 列中的文本。如果文本满足搜索自变量，则会返回 1，否则将返回 0。

```
SELECT AUTHOR,TITLE
       FROM DB2EXT.TEXTTAB
       WHERE CONTAINS(COMMENT, '"book"') = 1
```

在此示例中，在 COMMENT 列中搜索术语 book。

注意

要提高性能，添加限制性搜索条件很有用，例如：

```
SELECT AUTHOR,TITLE
       FROM db2ext.texttab
       WHERE CONTAINS(COMMENT, '"book"') = 1 AND PRICE < 20
```

搜索并返回找到的匹配项数目

使用 NUMBEROFMATCHES 函数来确定在每个文本文档中找到搜索值的频率。

```
SELECT AUTHOR,TITLE,
       NUMBEROFMATCHES(COMMENT, '"book"')
FROM DB2EXT.TEXTTAB
```

NUMBEROFMATCHES 将返回一个整数值。

搜索并返回找到的文本文档的分数

SCORE 是一个绝对值，它指示一个文档相对于其它找到的文档所满足搜索值的程度。该值指示在文档中找到的匹配数与文档大小的比例。在以下示例中，可以使用 SCORE 函数来获取找到的文档的分数：

```
WITH TEMPTABLE(docid,score)
  AS (SELECT docid,
            SCORE(COMMENT, '"book"')
      FROM DB2EXT.TEXTTAB)
SELECT *
  FROM TEMPTABLE
 WHERE score > 0
 ORDER BY score ASC
```

SCORE 将返回 0 到 1 之间的 DOUBLE 值。

注意

不能将 CONTAINS、SCORE 和 NUMBEROFMATCHES 搜索函数用于对视图创建的索引。

在分布式 DB2 环境中，SCORE 值是不同的：

- 在非分布式环境中，所有文档都在单个表中。SCORE 值是基于单个表以及文档与表中所有其它文档的关系。
- 在分布式 DB2 环境中，所有文档都位于不同的节点上。在建立索引期间，只有本地文档用来构建文本索引，这些索引都在每个节点本地。在这种情况下，SCORE 值基于文档与多个节点的仅其中一个节点中的所有文档之间的关系。

指定 SQL 搜索自变量

CONTAINS、NUMBEROFMATCHES 和 SCORE 函数都使用搜索自变量。本节使用 CONTAINS 函数来说明 DB2 Net Search Extender 函数中的搜索自变量的不同示例。

有关语法的描述，参考第 159 页的第 14 章，『搜索自变量的语法』。

按任意顺序搜索术语

一个搜索自变量中可以具有多个术语。组合几个搜索术语的一种方法是使用逗号将它们连接起来，如下所示：

```
SELECT AUTHOR,TITLE
      FROM DB2EXT.TEXTTAB
      WHERE CONTAINS(COMMENT,
        '("kid", "dinosaur")') = 1
```

这种格式的搜索自变量可以查找包含任何搜索术语的文本。在逻辑术语中，OR 运算符可连接搜索术语。

使用布尔运算符 AND 和 OR 进行搜索

可以使用布尔运算符 “&”（AND）和 “|”（OR）来将搜索术语与其它搜索术语组合在一起：

```
SELECT AUTHOR, TITLE
      FROM DB2EXT.TEXTTAB
      WHERE CONTAINS(COMMENT,
        '"author" | "pulitzer"') = 1
```

还可以使用布尔运算符来组合几个术语：

```
SELECT AUTHOR, TITLE
      FROM DB2EXT.TEXTTAB
      WHERE CONTAINS(COMMENT,
        '"author" | "pulitzer" & "book"') = 1
```

如果使用多个布尔运算符，则将按照从左到右的顺序对这些运算符进行求值。但是，逻辑 AND 运算符（&）优先于逻辑 OR 运算符（|）。可以在以下示例中了解此求值，它不包括圆括号：

```
"book" & "pulitzer" | year" & "author"
```

因此，Net Search Extender 按以下方式对布尔运算符进行求值：

```
("book" & "pulitzer") | (year & "author")
```

因此，要正确对布尔运算符进行求值，必须添加圆括号：

```
"book" & ("pulitzer" | year) & "author"
```

还可以将布尔运算符与使用逗号分隔符链接在一起的搜索术语进行组合。

```
("author", "pulitzer") & "book"
```

但是，在这种情况下，逗号将被解释为布尔运算符 OR：

```
("author" | "pulitzer") & "book"
```

有关附加信息，另见第 89 页的『使用布尔运算符 NOT 进行搜索』。

模糊搜索

“模糊”搜索将搜索与搜索术语拼写相似的词语。

```
SELECT AUTHOR, TITLE
FROM DB2EXT.TEXTTAB
WHERE CONTAINS(COMMENT,
                'fuzzy form of 80 "pullitzer") = 1
```

在此示例中，搜索可能会找到拼写错误的单词 pulitzer。

匹配级别（在该示例中为“80”）指定准确程度。当文档中可能具有拼写错误时，使用模糊搜索。当“光学字符识别”设备或语音输入创建文档时，通常会出现这种情况。使用 1 到 100（其中 100 是精确匹配，低于 80 是“模糊”匹配）。

注意

如果模糊搜索未提供适当的准确度，则使用字符屏蔽来搜索术语的部分。

搜索术语的部分（字符屏蔽）

屏蔽字符（或者称为“通配”字符）提供了一种使搜索更灵活的方法。它们通过增加由搜索找到的文本文档数来实现此目的。

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

- % 表示任何数目的任意字符。以下是在搜索术语中部将 % 用作屏蔽字符的一个示例：

```
SELECT AUTHOR, TITLE
FROM DB2EXT.TEXTTAB
WHERE CONTAINS(COMMENT, '"thr%er"') = 1
```

此搜索术语将查找包含词语“thriller”的文本文档。

- _ 表示搜索术语中的一个字符。以下示例还将查找包含词语“thriller”的文本文档。

```
SELECT AUTHOR, TITLE
FROM DB2EXT.TEXTTAB
WHERE CONTAINS(COMMENT, '"th_i1ler"') = 1
```

注意

要有节制地使用通配符，这是因为它们会极大地增加结果列表的大小，从而降低性能并且返回意外的搜索结果。

搜索已经包含屏蔽字符的术语

如果想要搜索包含 “%” 字符或 “_” 字符的术语，必须在该字符前添加通常所说的转义字符。然后，可以使用 ESCAPE 关键字来标识转义字符。

在以下示例中，转义字符为 "!"：

```
SELECT AUTHOR, TITLE
      FROM DB2EXT.TEXTTAB
      WHERE CONTAINS(COMMENT,
                     '"100!%" ESCAPE "!"') = 1
```

按固定顺序搜索术语

如果您搜索 “primary key”，则仅当这两个术语相邻并且按如下所示的顺序显示时才能够找到它们：

```
SELECT AUTHOR, TITLE
      FROM DB2EXT.TEXTTAB
      WHERE CONTAINS(COMMENT, '"primary key"') = 1
```

在同一句子或段落中搜索术语

以下是查找在其中搜索术语出现在同一句子中的文本文档的搜索自变量的示例：

```
SELECT AUTHOR, TITLE
      FROM DB2EXT.TEXTTAB
      WHERE CONTAINS(COMMENT,
                     '"web" IN SAME SENTENCE AS "disk"') = 1
```

还可以搜索同时出现的两个以上的词语：在下一个示例中，将搜索在同一段落中出现的两个词语：

```
SELECT AUTHOR, TITLE
      FROM DB2EXT.TEXTTAB
      WHERE CONTAINS(COMMENT,
                     '"computer" IN SAME PARAGRAPH AS "web"') = 1
```

在结构化文档的各节中搜索术语

以下是查找文本文档（在其中，搜索术语 IBM 出现在结构化文档的 H2 小节中）的搜索自变量的示例：

```
SELECT CATEGORY, DATE
      FROM DB2EXT.HTMLTAB
      WHERE CONTAINS(HTMLFILE,
                     'SECTIONS ("H2") "IBM"') = 1
```

注意，节名称是区分大小写的。确保模型文件和查询中的节名称是完全相同的。有关更多信息，参见第 97 页的第 9 章，『使用结构化文档』。

使用布尔运算符 NOT 进行搜索

可以使用布尔运算符 NOT 来从搜索中排除特定文本文档:

```
SELECT AUTHOR, TITLE
      FROM DB2EXT.TEXTTAB
     WHERE CONTAINS(COMMENT,
                    '("author", "pulitzer") & NOT "book"') = 1
```

此示例将从对 “author” 或 “pulitzer” 的搜索中排除包含术语 “book” 的任何文本文档。

同义字搜索

同义字搜索是 DB2 Net Search Extender 中的一项功能强大的搜索术语扩展功能。搜索的附加术语来自于您自己构建的同义字，因此，您可以直接控制术语。例如，搜索 “database” 可以找到诸如 “repository” 和 “DB2” 的术语。

对于您感兴趣的频繁搜索的特定区域使用这种类型的搜索，可以更有效地生成搜索结果。

有关更多信息以及如何构建同义字的描述，参见第 99 页的第 10 章，『使用同义字来扩展搜索术语』。以下示例将演示使用同义字扩展的语法。

此示例采用术语 “product” 并通过添加在同义字 “nsesamplethes” 中找到的此术语的所有关系来扩展它。在此示例中，已将 “marketing” 添加至搜索。

```
SELECT CATEGORY, DATE
      FROM DB2EXT.HTMLTAB
     WHERE CONTAINS(HTMLFILE,
                    'THESAURUS "nsesamplethes"
                    EXPAND RELATED
                    TERM OF "product"') = 1
```

下一个示例采用搜索术语 “product”。然后，使用该搜索术语的所有同义词来扩展搜索。

```
SELECT CATEGORY, DATE
      FROM DB2EXT.HTMLTAB
     WHERE CONTAINS(HTMLFILE,
                    'THESAURUS "nsesamplethes"
                    EXPAND SYNONYM
                    TERM OF "product"') = 1
```

数字属性搜索

对于存储在结构化文档中的数字属性，可以使用以下语法来进行搜索:

指定搜索自变量

```
SELECT AUTHOR, TITLE
      FROM DB2EXT.TEXTTAB
     WHERE CONTAINS(COMMENT,
                    'ATTRIBUTE "PRICE" between 9 and 20') = 1
```

注意，属性名是区分大小写的。确保模型文件和查询中的属性名是完全相同的。有关更多信息，参见第 97 页的第 9 章，『使用结构化文档』。

自由文本搜索

“自由文本搜索”是一种将搜索术语表示为自由格式文本的搜索。短语或句子用自然语言描述要搜索的主题。自由文本查询中词语的顺序是不相关的。但是，对于一组查询术语，在要搜索的文档中必须至少出现其中一个术语。

注意，不支持屏蔽自由文本自变量中的搜索字符串的字符或词语。

例如：

```
SELECT AUTHOR, TITLE, SCORE(COMMENT,
                             'IS ABOUT EN_US "something related to dinosaur"')
      FROM DB2EXT.TEXTTAB
     WHERE CONTAINS(COMMENT,
                    'IS ABOUT EN_US "something related to dinosaur"') = 1
```

通过将查询与 SCORE 搜索函数组合起来，搜索也将返回只包含词语 “something” 的文档。

附加搜索语法示例

要熟悉附加搜索语法示例，使用称为 `search` 的命令行处理器输入文件。这包含针对样本表运行的 DB2 Net Search Extender 搜索函数的示例。

要运行示例，使用以下语法：

```
db2 -tvf search
```

如果尚未创建表和索引，则运行下列其中一项：

- 在 UNIX 平台上：<instance_owner_home>/<sqllib>/samples/db2ext 目录中的 `nseample`。
- 在 Windows 平台上：<sqllib>/samples/db2ext 目录中的 `nseample (.bat)`。

使用存储过程搜索来搜索文本

当应用程序需要文本搜索结果的子集时，使用存储过程搜索界面，但是要采用一种高性能搜索方式。如果需要所有结果或者需要对大量文档建立索引，则不要使用存储过程。主要原因是已将用户表的多个部分复制到内存中，从而需要提供许多实际内存。

可以采用一种与游标导航相似的方式使用存储过程来首先请求 0 到 20 之间的结果，然后请求 21 到 40 的结果，依此类推。在具有此游标功能的情况下使用高速缓存（已在建立索引期间计算）会使搜索速度相当快，尤其是在不需要任何连接的情况下更是如此。

如果将使用存储过程，则务必考虑下列选项：

- 在分布式 DB2 环境中，必须在单个节点上为存储过程显式指定表空间，并在此节点上显式调用。
- 在 CREATE INDEX 期间已经指定了高速缓存搜索结果选项。
- 已经完全考虑了现在和将来的共享内存需求（可能涉及到增量更新）。有关进一步的信息，参见第 44 页的『创建存储过程搜索可以使用的文本索引』。
- 已经使用 db2text activate 命令激活了索引的高速缓存。

以下是存储过程搜索的一个示例：

```
db2 "call db2ext.textSearch('\"book\"','DB2EXT','COMMENT',0,2,1,1,?,?)"
```

第一个参数是搜索术语。该语法与 SQL 函数中的语法完全相同。然后，指定索引名和索引模式。如果尚未屏蔽该名称，则将它转换为大写。接着的两个数字提供了获得结果（以片段计）的可能性。接下来的两个整数值指定是否请求了分数和命中信息。最后两个值是输出值。

有关这些参数进一步的详细信息，参见第 179 页的第 16 章，『存储过程搜索函数』。

注意

如果请求较大的结果集，则需要用户表空间。如果没有提供任何表空间，则创建一个表空间。以下示例将在 UNIX 平台上创建一个表空间：

```
db2 "create user temporary tablespace tempts managed by system
      using ('/work/tempt.ts')"
```

在分布式 DB2 环境中，必须在存储过程的单个节点上为管理表显式指定表空间，并在此节点上显式调用。

使用“SQL 表值函数”来搜索文本

在下列情况下使用“SQL 表值函数”：实现了接口，但是不需要返回所有结果，或者没有使用存储过程接口所需要的所有实际内存。

提供了两个 SQL 表值函数，它们都称为 `db2ext.textsearch`。一个函数具有与 `db2ext.highlight` 函数配合使用的附加参数，您可以在没有此函数的情况下使用另一个函数。有关进一步的信息，参见『使用 HIGHLIGHT 函数』。

“SQL 表值函数”提供了与存储过程相同的游标界面来仅获取部分结果。但是，仍然需要将结果与用户表连接起来。可以在以下示例中了解这种情况：

```
db2 "select docid , author, score from TABLE(db2ext.textsearch('\\"book\\" ',
      'DB2EXT','COMMENT',3,2,cast(NULL as integer))) as t, db2ext.texttab u
      where u.docid = t.primkey"
```

下面是一些可能从“SQL 表值函数”返回的值：

```
--> primKey <single primary key type>
the primary key

--> score          DOUBLE
the score value of the found document

--> NbResults      INTEGER
the total number of found results (same value for all rows)

--> numberOfMatches INTEGER
the number of hits in the document
```

注意

注意，只允许存在单个主键列。有关这些参数进一步的详细信息，参见第 171 页的『DB2EXT.TEXTSEARCH』。

使用 HIGHLIGHT 函数

提供了两个 SQL 表值函数，它们都称为 `db2ext.textsearch`。要使用 `db2ext.highlight` 函数，必须使用带有附加 `numberOfHits` 和 `hitInformation` 参数的 `db2ext.textsearch` 函数。

在此示例中，会调用 `db2ext.highlight` 函数来显示整个文档，且不会突出显示由 `db2ext.textsearch` 函数找到的任何匹配项。

```
select p.docid,
db2ext.highlight(p.comment, t.hitinformation, ' WINDOW_NUMBER = 0 ')
      as highlight from DB2EXT.TEXTTAB p,
table (db2ext.textsearch('"bestseller" | "peacekeeping" | "soldiers"
```

```
| "attention"', 'DB2EXT', 'COMMENT', 0, 20,
      cast(NULL as INTEGER), 10)) t
where p.docid = t.primkey and p.docid = 2
```

搜索自变量将返回以下结果:

DOCID HIGHLIGHT

```
2      A New York Times bestseller about peacekeeping soldiers called
      "Keepers" who devise a shocking scheme to get the worlds
      attention after their tour of duty ends.
```

1 record(s) selected.

注意

在所有 db2ext.highlight 示例中, 表函数 db2ext.textsearch 搜索下列词语中的任何一项: “bestseller”、“peacekeeping”、“soldiers”或“attention”。

在此示例中, 会调用 db2ext.highlight 函数来显示整个文档, 并且突出显示由 db2ext.textsearch 函数找到的所有匹配项。

```
select p.docid,
db2ext.highlight(p.comment, t.hitinformation, ' WINDOW_NUMBER = 0,
      TAGS = ("<bf>", "</bf>" ) ') as highlight from DB2EXT.TEXTTAB p,
table (db2ext.textsearch('"bestseller" | "peacekeeping" | "soldiers"
      | "attention"', 'DB2EXT', 'COMMENT', 0, 20,
      cast(NULL as INTEGER), 10)) t
where p.docid = t.primkey and p.docid = 2
```

搜索自变量将返回以下结果:

DOCID HIGHLIGHT

```
2      A New York Times <bf>bestseller</bf> about <bf>peacekeeping</bf>
<bf>soldiers</bf> called "Keepers" who devise a shocking scheme to
      get the worlds <bf>attention</bf> after their tour of duty ends.
```

1 record(s) selected.

在此示例中, 会调用 db2ext.highlight 函数来最多显示文档的 10 个部分(窗口)。每个窗口大小为 24, 在匹配项的每一边大约具有 12 个字节的数据。此外, 将突出显示由 db2.textsearch 表函数找到的匹配项。

```
select p.docid,
db2ext.highlight(p.comment, t.hitinformation, ' WINDOW_NUMBER = 10,
      WINDOW_SIZE = 24, TAGS = ("<bf>", "</bf>" ) ') as highlight
from DB2EXT.TEXTTAB p,
table (db2ext.textsearch('"bestseller" | "peacekeeping" | "soldiers"
```

附加搜索语法示例

```
| "attention", 'DB2EXT', 'COMMENT', 0, 20,  
      cast(NULL as INTEGER), 10)) t  
where p.docid = t.primkey and p.docid = 2
```

搜索自变量将返回以下结果:

DOCID HIGHLIGHT

```
2      York Times <bf>bestseller</bf> about <bf>peacekeeping</bf> ...  
      <bf>peacekeeping</bf> <bf>soldiers</bf> called "Keepers" ... the  
      worlds <bf>attention</bf> after their
```

1 record(s) selected.

找到的第一个匹配项为 **<bf>bestseller</bf>**，此匹配项将确定第一个窗口。第二个匹配项 **<bf>peacekeeping</bf>** 与第一个匹配项只相隔 8 个字节，会将它完全放置在第一个窗口中。第三个匹配项 **<bf>soldiers</bf>** 在第一个窗口外边，它将确定新窗口。由于第二个匹配项 **<bf>peacekeeping</bf>** 只与 **<bf>soldiers</bf>** 匹配项的左边相隔 2 个字节，因此，也会将它放置在第二个窗口中，并进行突出显示。第四个匹配项 **<bf>attention</bf>** 在第二个窗口外部，所以它将确定新窗口。由于此窗口中没有包含任何先前的或附加的匹配项，因此，该窗口中只包含匹配项附近的数据。

另外，因为没有指定任何 WINDOW_SEPARATOR，所以采用缺省窗口分隔符“...”来分隔这三个文档窗口。

注意

为了确保在使用 db2ext.highlight 函数时具有高性能，用户应限制 db2ext.textsearch 表值函数中的搜索结果。

有关这些参数进一步的详细信息，参见第 175 页的『DB2EXT.HIGHLIGHT』。

搜索多列

在需要对多列创建文本索引的情况下，最简单的方法是使用 SQL 标量函数并且组合针对这些列的搜索。可在以下示例中看到这一情况:

```
SELECT AUTHOR,TITLE  
      FROM DB2EXT.TEXTTAB  
      WHERE CONTAINS(COMMENT,  
      'book')=1 and CONTAINS(AUTHOR,'Mike')=1
```

对于表值函数，这会更困难，原因是由于性能原因您需要使用并集。使用表值函数的另一种可能是使用视图并且组合视图列中的表列以创建文本索引。这样可以避免出现两个独立的文本搜索调用。

组合文本列可提高性能。但是，这主要取决于个别搜索需求。

性能注意事项

要在搜索期间提高性能，应考虑下列问题：

- 何时在 SQL 内进行搜索：
 - 如果注意到性能降低了，则使用 `explain` 语句来检查“DB2 优化器”的处理方案。
 - 参数搜索的搜索速度可以更快，尤其是当使用其它搜索谓词来减少结果大小时。
 - 如果不需要所有结果，则使用结果限制关键字。
- 何时使用存储过程进行搜索：
 - 由于已将指定的高速缓存表表达式从数据库复制到内存中，因此，应确保工作站具有足够的内存可用于此数据。如果内存不足，则会使用调页空间，这将降低搜索性能。

注意

有关最新的性能提示，请访问以下 DB2 Net Search Extender Web 站点：
www.ibm.com/software/data/db2/extenders/netsearch/index.html

第 9 章 使用结构化文档

DB2 Net Search Extender 允许您为文本或数字字段（例如，结构化文档中的标题、作者或描述）建立索引和搜索它们。文档可以采用 XML、Outside-In 或 HTML 格式，也可以包含用户定义标记（GPP）。

在文档模型中使用标记（markup tag）及其字段名来定义为文档中的哪些字段建立了索引，因此，可以搜索这些字段。可以在对字段的查询中使用该字段的名称（也称为节名）。

为了能够在这些字段中进行搜索，当创建包含文档的文本索引时必须指定 `FORMAT AND MODEL FILE`。有关进一步的信息，参见第 130 页的『CREATE INDEX』。

有关创建和定义文档模型的更多信息，参见第 185 页的第 17 章，『结构化文档支持』。

有关文档模型语法，参见第 253 页的附录 G，『文档模型参考』。

第 10 章 使用同义字来扩展搜索术语

不仅可以通过搜索特定搜索术语还可以搜索与其相关的术语来扩大查询范围。可以通过使用 Net Search Extender 的根据同义字查找并抽取相关搜索术语的功能来使此过程自动化。同义字是语义上相关的术语的受控词汇表，它通常涵盖特定主题区。

DB2 Net Search Extender 允许您根据先前创建的同义字添加附加术语来扩展搜索术语。参考第 159 页的第 14 章，『搜索自变量的语法』以查明如何在查询中使用同义字扩展。

为了创建同义字以将其用于搜索应用程序，此时需要必须编译为内部格式的同义字定义文件，即同义字典。

本章描述以下内容：

- **同义字的结构**

同义字的结构类似根据关系链接在一起的节点网络。本节描述 Net Search Extender 的预定义关系以及如何定义您自己的关系。

- **创建和编译同义字**

以下是同义字定义文件语法的描述以及用来将该文件编译为同义字典的工具的描述。

同义字的结构

同义字的结构类似根据关系链接在一起的节点网络。Net Search Extender 在同义字中查找术语的方法是从该术语开始，然后沿着通过该术语关系的路径发送在该过程中找到的术语。

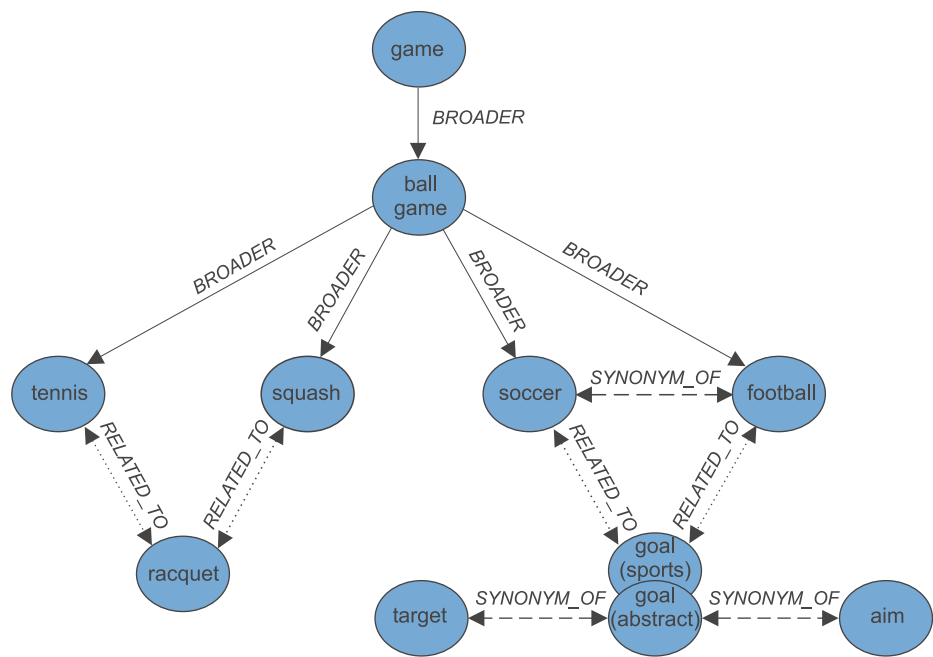


图 28. 同义字结构的示例

同义字条目是根据关系连接的。关系名（例如，BROADER）允许您将扩展限制为关系层次结构中的某些命名行。某些关系是双向的，其它是非双向的；例如，BROADER 是非双向关系的名称。

预定义同义字关系

以下是 Net Search Extender 中的预定义关系：

- 关联关系

关联关系是两个术语之间的双向关系，这两个术语并不表示同一概念，但是彼此相关。

预定义关联关系：RELATED_TO

示例：

```
tennis RELATED_TO racket
football RELATED_TO goal (sports)
```

- 同义词关系

同义词关系是两个术语之间的双向关系，这两个术语有相同或相似的含义并且可以彼此替代。例如，此关系可以用在术语与其缩写之间。

预定义同义词关系：SYNONYM_OF

示例：

```
spot SYNONYM_OF stain
US SYNONYM_OF United States
```

第 100 页的图 28 显示相同同义字中的两个 goal 术语。一个使用注释 (sports) 指定，另一个使用注释 (abstract) 指定。即使术语拼写相同，同义词关系仍可连接不同词组。可以通过在定义同义字时使用不同的关系来形成这一点。

• 层次结构关系

层次结构关系是在两个术语之间的非双向关系，一个术语比另一个术语具有更广泛（更普遍）的意义。根据关系的方向，可以使用该关系来查找更专业或更普遍的术语。

预定义层次结构关系：

- NARROWER 用来形成缩小关系

NARROWER 关系用来形成更专业的术语序列。随着搜索关系的范围越来越小，术语变得越具体。例如，如果沿着 NARROWER 关系查找术语 house，则结果在越来越专业的术语的列表中可能包括 skyscraper palace church chapel cathedral 等等。

- BROWDER 用来形成扩大关系。

BROWDER 关系用来形成越来越普遍的术语的序列。随着这种关系的搜索范围越来越大，术语将变得越不具体。例如，如果沿着 BROWDER 关系查找术语 house，则结果在越来越普遍的术语列表中可能包括 building construction object 等等。

定义您自己的关系

Net Search Extender 允许您定义自己的 RELATED_TO、NARROWER 和 BROWDER 同义字关系。因为每个关系名必须是唯一的，所以必须通过添加唯一编号来限定这种关系名，如：RELATED_TO(42)。

可以使用相同关系编号来定义不同类型的关系，如 NARROWER(42)。编号 0 用来指 Net Search Extender 的预定义关系。

创建和编译同义字

执行下列步骤来创建可以由 Net Search Extender 功能使用的同义字：

1. 创建同义字定义文件。
2. 将定义文件编译为同义字典。

创建同义字定义文件

要创建您自己的同义字，第一步是使用文本编辑器在定义文件中定义它的内容。

限制。文件名（包括扩展名）的长度一定不能超过 256 个字符。同一目录中可以有几个同义字，但建议每个同义字具有独立的目录。

提供了样本英语同义字定义文件 `nsesamplethes.def`。Windows 系统的同义字目录为:

```
<sqllib>\db2ext\thes
```

在 UNIX 系统上，同义字目录为:

```
<instance_owner_home>/sqllib/db2ext/thes
```

以下是该文件中的前几个定义组:

```
:WORDS
    accounting
    .RELATED_TO account checking
    .RELATED_TO sale management
    .SYNONYM_OF account
    .SYNONYM_OF accountant

:WORDS
    acoustics
    .RELATED_TO signal processing

:WORDS
    aeronautical equipment
    .SYNONYM_OF turbocharger
    .SYNONYM_OF undercarriage

:WORDS
    advertising
    .RELATED_TO sale promotion
    .SYNONYM_OF advertisement
:
:
:
```

图 29. 样本同义字定义文件的片断

有关每个定义组的语法，参见第 197 页的第 18 章，『同义字支持』。

必须将每个成员写至单个行。每个相关联的术语前必须加上关系名。如果成员术语彼此相关，则指定成员关系。

成员术语和相关联术语的长度限制为 64 个字符。同一字母的单字节字符和双字节字符被视作相同。不区分大写和小写字母。术语可以包含空白字符，并且可以使用单字节字符句点 “.” 或冒号 “:”。

用户定义关系都是以关联类型为基础的。它们由 1 与 128 之间的唯一数字标识。

将定义文件编译为同义字典

要编译同义字定义文件，运行 `db2extth` 命令。有关命令语法，参见第 149 页的『DB2EXTTH（实用程序）』。

要在分区环境中使用同义字典，确保所有物理节点都可存取创建的文件。

提示

参见第 277 页的附录 M，『同义字工具返回的消息』。

使用同义字

第 2 部分 参考

第 11 章 实例所有者的管理命令

本章描述实例所有者的管理命令的语法。实例所有者管理包括检查 DB2 Net Search Extender 锁定和更新服务的状态以及启动和停止这些服务。

有关附加信息，参见第 29 页的第 5 章，『Net Search Extender 实例服务』。

这些命令是 DB2TEXT 命令的变体，并且允许管理特定于 DB2 实例的 “DB2 Net Search Extender 服务”。

命令	用途	页
CONTROL	列示和删除全文本索引锁定。还将列示高速缓存状态。	108
START	启动 DB2 Net Search Extender 实例服务。	110
STOP	停止 DB2 Net Search Extender 实例服务。	111

CONTROL

此命令允许您列示和删除由 “Net Search Extender 实例服务” 管理的全文本索引锁定。如果锁定和更新服务正在运行，可以查看它们的状态以及有关已激活高速缓存的信息。

在分布式 DB2 环境中，这仅影响当前分区。用户负责为期望的分区调用 DB2 命令 db2_all。

授权

必须作为 DB2 实例所有者在服务器上运行此命令。

命令语法



命令参数

CLEAR

使用 **CLEAR** 来强制清除一组锁定。在检查导致锁定问题的原因后谨慎地使用此命令。

LIST

使用 **LIST** 来获取关于对特定索引或数据库持有的当前锁定的信息。如果有更新锁定，则可以获取关于已处理文档的信息。

还可以使用 **LIST** 查看在更新过程中更新的文档数。注意这只适用于锁定持有索引的时候。

set-of-locks

仅在指定数据库或索引中使用锁定。

SHOW CACHE STATUS FOR

显示指定索引的高速缓存表的激活状态。可以是 “Not Activated” 或 “Currently Activated”。如果已激活高速缓存，则它将显示关于高速缓存使用情况的详细信息。例如，最大高速缓存大小（以兆字节计）、要插入的最大文档数和保留在高速缓存表中的空间（以千字节计）。

STATUS

通过使用 STATUS 关键字，该命令显示锁定和更新 “Net Search Extender 实例服务” 是否已启动并正在运行。

DATABASE database name

正在使用的服务器上的数据库的名称。

INDEX index-schema.index-name

目前正在使用的文本索引的模式和名称。它是在 CREATE INDEX 命令中指定的。

使用法

当管理命令错误消息指示存在锁定问题时，确保没有任何冲突任务正在运行。例如，在 UPDATE 命令运行时尝试运行 ALTER 命令。然后释放索引的所有锁定。

对增量索引更新使用 SHOW CACHE STATUS 来检查指定内存大小是否仍然大到足以保存下一次更新期间的所有更新信息，或检查是否已经完成激活。

START

此命令将启动守护程序，它控制 DB2 服务器上的全文本索引的锁定以及全文本索引的自动更新。

注意

由于该命令不会对索引激活任何临时高速缓存表，所以在使用存储过程进行搜索时需要个别 `ACTIVATE CACHE` 命令。

授权

必须作为 DB2 实例所有者在服务器或分布式 DB2 环境中的任何一个服务器上运行此命令。

命令语法

►—START—◄◄

命令参数

无。

使用法

在 Windows 上，该命令将启动 `db2ext-<InstanceName>` 服务。还可以使用常规 Window 方法启动此命令。

对于全文本索引的锁定，可以修改配置文件以满足您的需求。有关进一步的信息，参见第 29 页的『锁定服务』。

STOP

此命令将停止 Net Search Extender 的锁定和更新服务。

授权

必须作为 DB2 实例所有者在服务器或分布式 DB2 环境中的任何一个服务器上运行此命令。

命令语法

```
➤—STOP—┐
           └—FORCE—┘
```

命令参数

FORCE

即使进程现在持有锁定或对任何索引激活了高速缓存表，仍然会停止服务。如果不指定 FORCE，则该命令在这些情况下将会失败。

使用法

停止“Net Search Extender 实例服务”将不会允许继续使用特定的 Net Search Extender 命令。当重新启动服务时，必须再次激活临时高速缓存（如果先前将已激活的高速缓存与索引配合使用的话）。

注意，已激活高速缓存或运行 Net Search Extender 命令将不会停止服务。

对于全文本索引的锁定，可以修改配置文件以满足您的需求。有关进一步的信息，参见第 29 页的『锁定服务』。

STOP 命令

第 12 章 数据库管理员的管理命令

本章描述数据库管理员的管理命令的语法。数据库管理包括设置数据库以供 DB2 Net Search Extender 使用然后禁用此设置。

第 35 页的第 6 章，『创建和维护文本索引』描述如何使用这些命令。

只有 ENABLE DATABASE 和 DISABLE DATABASE 命令才是 DB2TEXT 命令的变体，即使所有这些命令都允许在数据库级别进行管理。

命令	用途	页
ENABLE DATABASE	使当前数据库能够创建全文本索引。	114
DISABLE DATABASE	复位 DB2 Net Search Extender 为数据库完成的准备工作。	116
DB2EXTDL（实用程序）	缺省 UDF，用来检索数据链路文本列的内容。	118
DB2EXTHL（实用程序）	缺省 UDF 使用 100 KB 的文档并返回 200 KB 的 CLOB。	119

提示

如果没有任何数据库连接被指定为 db2text 命令的一部分，则 db2text 可执行文件将会建立与在环境变量 DB2DBDFT 中指定的缺省数据库的隐式连接。

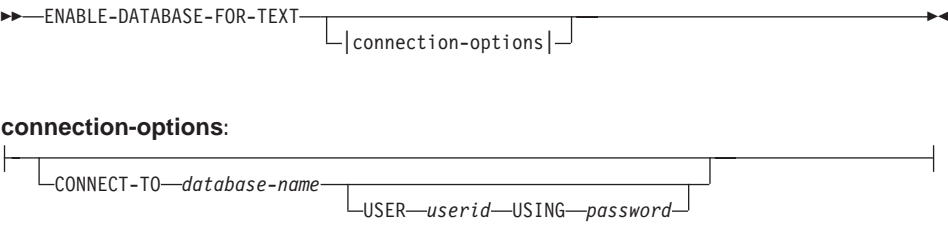
ENABLE DATABASE

此命令使数据库能够对文本列创建和利用全文索引。

授权

必须作为数据库管理员运行此命令才能启用数据库。这要求您具有 SYSADM 权限以便能够将 DBADM 授予 DB2 实例所有者。

命令语法



命令参数

CONNECT TO database-name

作为此命令的目标的数据库的名称。如果设置了 DB2DBDFT 并且用户使用具有必要的 DB2 权限的用户标识运行该命令，则可以省略此参数。

USER userid USING password

使用 password 和 userid 来连接至该数据库。

使用法

此命令准备已连接的数据库供 DB2 Net Search Extender 使用。经过此必要步骤，才可对数据库中的表 / 列创建 DB2 Net Search Extender 索引。

可以通过使用 DB2EXT.DBDEFAULTS 目录视图来查看运行该命令后建立的数据库缺省值。

对数据库的更改

此命令将 DBADM 权限授予与已启用数据库的 DB2 实例相关联的 DB2 实例所有者。

ENABLE DATABASE 命令在 DB2EXT 模式中创建各种数据库对象，如 DB2 Net Search Extender 目录、UDF 和存储过程。在运行该命令后，下列目录视图可用：

```
db2ext.dbdefaults  
db2ext.textindexes  
db2ext.textindexformats  
db2ext.indexconfiguration  
db2ext.proxyinformation
```

注意，由于向下兼容，DB2 Text Information Extender 视图也是可用的。有关进一步的信息，参见第 211 页的附录 C，『Net Search Extender 信息目录』。

还要注意，上面的表位于称为 IBMDEFAULTGROUP 的数据库的缺省表空间中。它分布在 db2nodes.cfg 中定义的所有节点上。

对文件系统的更改

无。

注意

如果在数据库中定义了任何文本索引，则禁用数据库将失败。建议依次除去这些索引，然后检查是否有任何问题发生。如果将禁用数据库用于文本强制命令，则它只保证除去数据库中的 Net Search Extender 目录表。

但是，如果一些索引未能完全删除，则可能仍然有一些资源需要手工清除。这些资源包括：

- 索引、工作和高速缓存目录中的文件
- ctedem.dat 中的调度程序条目
- 其中索引是使用复制捕获选项创建的，而远程数据库的表中的 IBMSNAP_SIGNAL、IBMSNAP_PRUNE_SET 和 IBMSNAP_PRUNCNTL 条目必须手工删除。使用 APPLY_QUAL="NSE" || <instance name> and TARGET_SERVER= <database name> 命令很容易标识这些条目。

在以下示例中，实例是 DB2，而数据库是 SAMPLE。

```
DELETE FROM <ccSchema>.IBMSNAP_SIGNAL
WHERE SIGNAL_INPUT_IN IN
      (SELECT MAP_ID FROM <ccSchema>.IBMSNAP_PRUNCNTL
       WHERE APPLY_QUAL= 'NSEDB2' AND TARGET_SERVER= 'SAMPLE');

DELETE FROM <ccSchema>.IBMSNAP_PRUNCNTL
WHERE APPLY_QUAL= 'NSEDB2' AND TARGET_SERVER= 'SAMPLE';

DELETE FROM <ccSchema>.IBMSNAP_PRUNE_SET
WHERE APPLY_QUAL= 'NSEDB2' AND TARGET_SERVER= 'SAMPLE';
```

对数据库的更改

将删除下列在数据库中用来启用 DB2 Net Search Extender 的修改：

- 数据库中的 DB2 Net Search Extender 目录视图。
- 由 DB2 Net Search Extender 创建的所有数据库对象。

对文件系统和共享内存的更改

如果使用 FORCE 选项，将删除索引文件。

如果使用 FORCE 选项，将删除作为索引的任何已激活高速缓存的高速缓存。有关进一步的信息，参见第 147 页的『DROP INDEX』。

DB2EXTDL (实用程序)

缺省情况下, 检索 Data Link 文本列的内容的 UDF 返回 100 KB BLOB。根据 Data Link 引用的数据库中最大文档的大小, 可以增加或减少此值。

授权

必须作为数据库管理员运行此命令才能启用数据库。这要求您具有 SYSADM 权限以便能够将 DBADM 授予 DB2 实例所有者。

命令语法

►►db2extdl—*new-result-size*—————►◄

命令参数

new-result-size

检索 Data Link 内容的 UDF 的新结果大小 (以千字节计)。它是小于 2097152 的正整数。

DB2EXTHL (实用程序)

缺省情况下, 突出显示 UDF 将最大大小为 100 KB 的文档作为输入并返回 200 KB CLOB。根据数据库中最大文档的大小, 可以将输入值增大为最大大小 1 GB。

授权

必须作为数据库管理员运行此命令才能启用数据库。这要求您具有 SYSADM 权限以便能够将 DBADM 授予 DB2 实例所有者。

命令语法

►►—db2exthl—*new-highlight-input-size*—————►◄

命令参数

new-highlight-input-size

突出显示 UDF 的新结果大小 (以千字节计)。它是小于 1048576 的正整数。

第 13 章 文本表所有者的管理命令

本章描述文本表所有者的管理命令的语法。

第 35 页的第 6 章，『创建和维护文本索引』描述如何使用这些命令。

这些命令是 DB2TEXT 命令的变体。这些命令允许表的所有者创建和处理表列的全文本索引。

命令	用途	页
ACTIVATE CACHE	激活高速缓存以便可以执行使用存储过程的搜索操作	122
ALTER INDEX	更改索引的特征	124
CLEAR EVENTS	从在索引更新期间使用的索引事件表中删除索引事件	128
CREATE INDEX	创建全文本索引	130
DEACTIVATE CACHE	取消激活高速缓存以使使用存储过程的搜索操作不再可行	145
DB2EXTTH（实用程序）	编译同义字定义文件	149
DROP INDEX	删除文本列的全文本索引	147
ENABLE DATABASE	启用当前数据库以创建全文本索引	114
UPDATE INDEX	启动基于文本列的当前内容的建立索引过程	151
HELP	显示 DB2TEXT 命令选项的列表	155
COPYRIGHT	显示 Net Search Extender 产品和版权信息	157

提示

如果没有任何数据库连接被指定为 db2text 命令的部分，则 db2text 可执行文件将会建立与在环境变量 DB2DBDFT 中指定的缺省数据库的隐式连接。

ACTIVATE CACHE

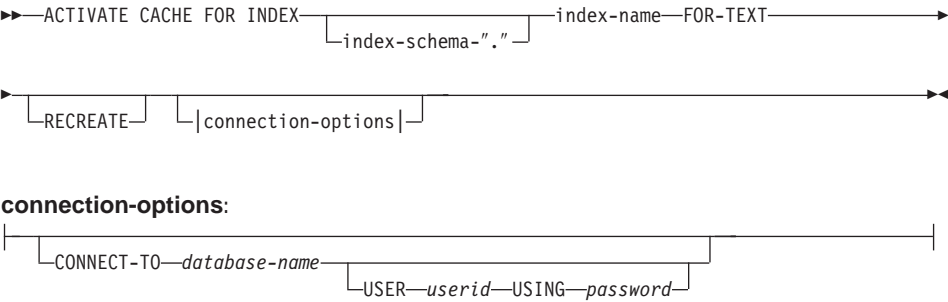
此命令激活 DB2 用户表或持久高速缓存中已高速缓存的表。在完成之后，就可以执行使用存储过程的搜索操作。有关进一步的信息，参见第 179 页的第 16 章，『存储过程搜索函数』。

仅当索引是使用 CACHE TABLE 选项创建的时候此命令才可用。有关进一步的信息，参见第 130 页的『CREATE INDEX』。

授权

根据 DB2 目录视图，此命令中的用户标识必须对为其创建全文本索引的表具有 CONTROL 特权。

命令语法



命令参数

index-schema

文本索引的模式，如在 CREATE INDEX 命令中指定的那样。如果没有指定任何模式，则使用 DB2 连接的用户标识。

index-name

文本索引的名称，如在 CREATE INDEX 命令中指定的那样。

RECREATE

仅适用于使用持久高速缓存的索引；会删除现有高速缓存。如果完成了没有激活的更新，则会从数据库中自动重构持久高速缓存。

CONNECT TO database-name

作为此命令的目标的数据库的名称。如果设置了 DB2DBDFT 并且用户正在服务器上运行命令，则可以省略此参数。注意，用户标识必须具有必需的 DB2 权限。

USER userid USING password

使用 password 和 userid 来连接至数据库。如果未指定它们，则会尝试使用当前用户标识而不带密码来建立连接。

使用法

如果正在对索引运行下列命令之一，则不能发出该命令：

- UPDATE INDEX
- ALTER INDEX
- DROP INDEX
- CLEAR EVENTS
- DEACTIVATE CACHE

注意

激活高速缓存的表可能需要从头开始重新创建它，即使使用的是持久高速缓存也是这样。如果在取消激活持久高速缓存的同时执行更新操作，则会发生这种情况。

用来构建高速缓存的内存量是根据当前文档数和结果列大小来动态计算的。使用 PCTFREE 值来将计算的最小内存量乘以因子 $100/(100-PCTFREE)$ 。PCTFREE 值是在 CREATE 或 ALTER INDEX 命令中指定的。

因此，PCTFREE 描述激活了高速缓存时保留给插入操作的已分配高速缓存的百分比。注意，对于每个 ACTIVATE CACHE 命令，都会重新估计实际内存大小。

对文件系统的更改

创建了用于实现持久高速缓存的文件。

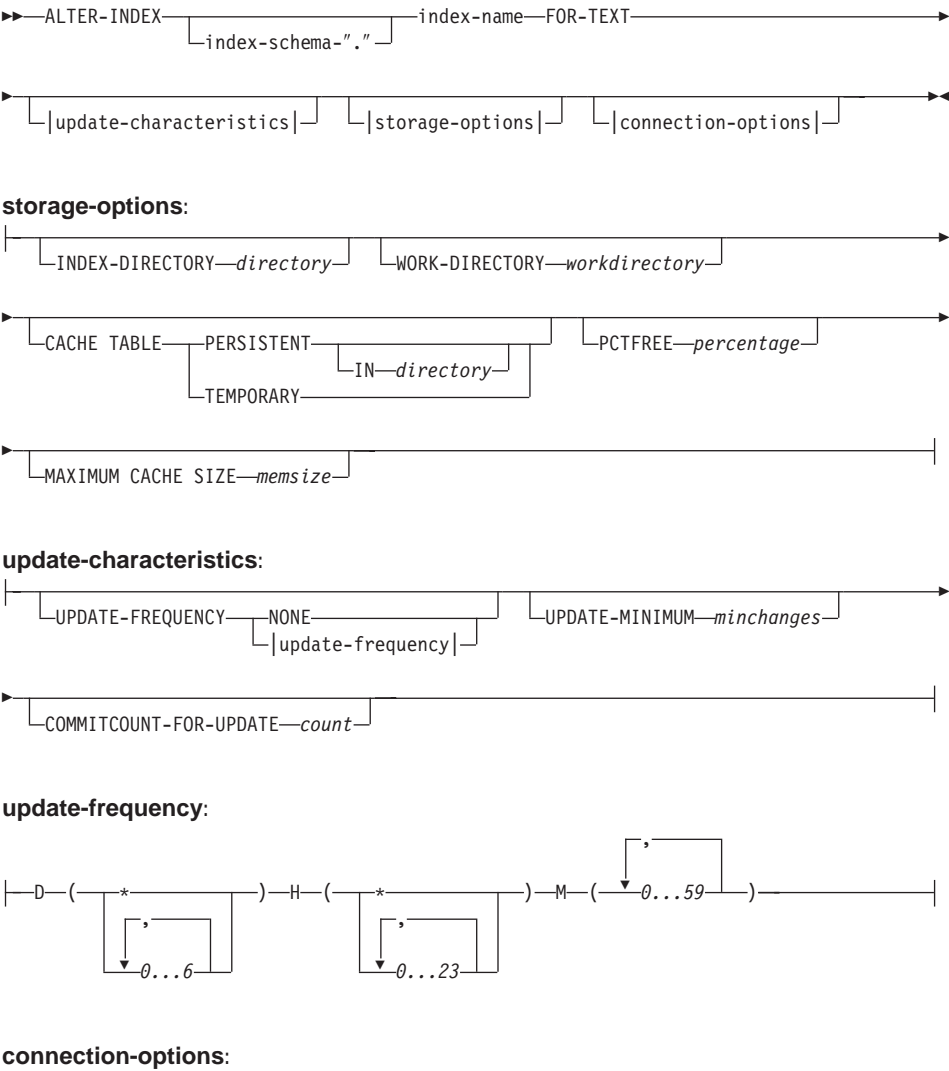
ALTER INDEX

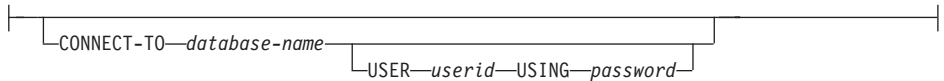
该命令将更改全文本索引的特征，例如，更新选项和存储器选项。

授权

根据 DB2 目录视图，此命令中的用户标识必须对为其创建全文本索引的表必须具有 CONTROL 特权。

命令语法





命令参数

index-schema

文本索引的模式，如在 CREATE INDEX 命令中指定的那样。如果没有指定任何模式，则使用 DB2 连接的用户标识。

index-name

文本索引的名称，如在 CREATE INDEX 命令中指定的那样。

INDEX DIRECTORY directory

用来存储文本索引的目录路径。由于该目录将包含索引数据，因此应确保该目录对 DB2 实例所有者用户标识具有读 / 写和运行许可权。

注意，在分布式 DB2 环境中，每个节点上都必须存在此目录。在目录下创建了一个子目录 NODE<nr> 以区分服务器的逻辑节点上的索引。将删除先前索引目录中的任何索引文件。

WORK DIRECTORY workdirectory

在搜索和管理操作期间用来存储临时文件。可以独立地更改新的索引目录的独立工作目录。

如果该目录不存在，则会为 DB2 实例所有者用户标识创建它。如果该目录存在，则确保在 UNIX 平台上实例所有者对该目录具有读 / 写许可权。

注意，在分布式 DB2 环境中，每个节点上都必须存在此目录。在目录下创建了一个子目录 NODE<nr> 以区分服务器的逻辑节点上的索引。将删除先前索引目录中的任何临时索引文件。

CACHE TABLE PERSISTENT IN directory

指定在取消激活或系统重新引导之后，CREATE INDEX 中已高速缓存的表就是持久的了。在任何一种情况下，这都允许快速执行 ACTIVATE CACHE。持久高速缓存存储在指定的目录中。

先前创建的持久高速缓存将移至新位置。此位置总是需要已取消激活的索引。

CACHE TABLE TEMPORARY

指定高速缓存的结果表现在是临时的，并且已经删除了任何先前存在的持久高速缓存。注意，此更改需要已取消激活的索引。

MAXIMUM CACHE SIZE memsize

指定在 ACTIVATE CACHE 期间要构建的高速缓存表的新的最大大小。以兆字节为单位将 memsize 参数指定为正整数。

如果该整数太小，则 `ACTIVATE CACHE` 命令将失败。实际高速缓存大小是在执行 `ACTIVATE CACHE` 命令期间计算的。此更改需要已取消激活的索引。

PCTFREE percentage

指定保留的用于附加文档的高速缓存的百分比。该百分比必须是小于 100 并且大于或等于 0 的整数。注意，会删除先前的持久高速缓存，并且此更改需要已取消激活的索引。参见第 122 页的『`ACTIVATE CACHE`』。

UPDATE FREQUENCY

通过使用下列参数，索引更新步骤可确定何时进行更新：

- **D.** 在星期几更新索引：*（每天）或 0..6（0 = 星期天）
- **H.** 在几点钟更新索引：*（每小时）或 0..23
- **M.** 在多少分钟更新索引：0..59
- **NONE.** 不再进行进一步的索引更新。对于不再进一步更改的文本列，就可以使用此参数。

如果不指定 `UPDATE FREQUENCY` 关键字，则频率设置将保持不变。

UPDATE MINIMUM minchanges

在对索引进行增量更新之前允许对文本文档所做的最小更改数。如果不指定 `UPDATE MINIMUM` 关键字，则不会更改该设置。

注意，仅当索引不是使用 `RECREATE ON UPDATE` 选项创建时才能更改 `UPDATE MINIMUM`。

COMMITCOUNT FOR UPDATE count

对于更新处理，可以指定落实计数。有关进一步的信息，参见第 151 页的『`UPDATE INDEX`』。这同时适用于 `UPDATE` 命令和 `UPDATE FREQUENCY` 规范，它将调度更新处理。

注意，仅当索引不是使用 `RECREATE ON UPDATE` 选项创建时才能更改 `COMMITCOUNT`。

还要注意，如果是使用 `REPLICATION` 选项来创建索引的，则不能更改 `COMMITCOUNT`。

CONNECT TO database-name

作为此命令的目标的数据库的名称。如果设置了 `DB2DBDFT` 并且用户正在服务器上运行命令，则可以省略此参数。注意，用户标识必须具有必需的 `DB2` 权限。

USER userid USING password

使用 `password` 和 `userid` 来连接至数据库。如果未指定它们，则会尝试使用当前用户标识而不带密码来建立连接。

使用法

如果正在对索引运行下列命令之一，则不能发出该命令：

- ALTER INDEX
- CLEAR EVENTS
- ACTIVATE CACHE
- DROP INDEX
- UPDATE INDEX
- DEACTIVATE CACHE

在分布式 DB2 环境中，仅在具有单个节点的表空间上才允许具有高速缓存选项的文本索引。

对数据库的更改

更改 DB2 Net Search Extender 目录视图。

对文件系统的更改

- 在索引中创建 NODE<nr> 子目录和工作目录
- 移动索引文件
- 创建持久高速缓存目录
- 移动持久高速缓存文件

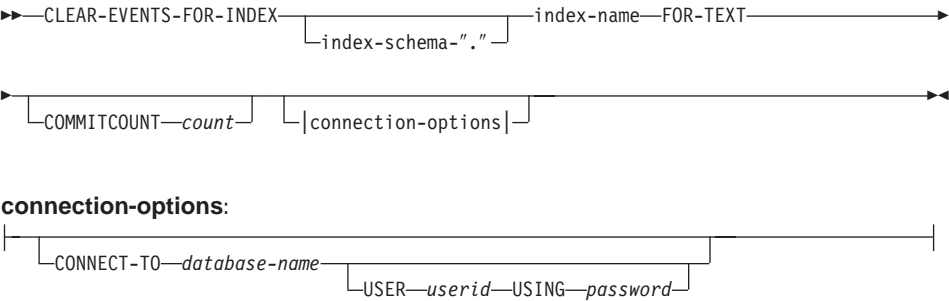
CLEAR EVENTS

此命令将从索引的事件视图中删除建立索引事件。将事件视图用于管理。可在 DB2EXT.TEXTINDEXES 视图的 EVENTVIEWNAME 列中找到事件视图的名称。

授权

根据 DB2 目录视图，此命令中的用户标识必须对为其创建全文本索引的表必须具有 CONTROL 特权。

命令语法



命令参数

index-schema

文本索引的模式，如在 CREATE INDEX 命令中指定的那样。如果没有指定任何模式，则使用 DB2 连接的用户标识。

index-name

文本索引的名称，如在 CREATE INDEX 命令中指定的那样。

COMMITCOUNT count

大于等于 0 的 INTEGER 值显示由 DB2 在一个事务中删除的行数。

CONNECT TO database-name

作为此命令的目标的数据库的名称。如果设置了 DB2DBDFT 并且用户正在服务器上运行命令，则可以省略此参数。注意，用户标识必须具有必需的 DB2 权限。

USER userid USING password

使用 password 和 userid 来连接至数据库。如果未指定它们，则会尝试使用当前用户标识而不带密码来建立连接。

使用法

当在 CREATE 或 ALTER INDEX 命令中使用 UPDATE FREQUENCY 选项来调度常规更新时，就会定期检查事件表。在检查了事件的原因并除去了错误源之后，使用 CLEAR EVENTS 来清理事件表。

尝试确保表和索引中的文本列的内容之间的一致性，尤其是在对文档重新建立索引时。

如果正在对索引运行下列命令之一，则不能发出该命令：

- UPDATE INDEX
- ALTER INDEX
- ACTIVATE CACHE
- DEACTIVATE CACHE
- DROP INDEX

CREATE INDEX

此命令对文本列创建全文索引以供在 DB2 Net Search Extender 全文本查询中使用。

在分布式 DB2 环境中，全文索引是对在其上定义用户表的表空间的每个分区创建的。不允许对表空间的分布进行后续更改，这将导致在执行管理命令和搜索过程中产生意外行为。

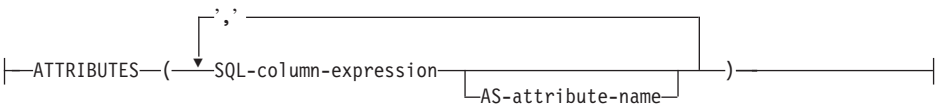
授权

根据 DB2 目录视图，此命令中的用户标识必须对在表中创建全文索引的表具有 CONTROL 特权。

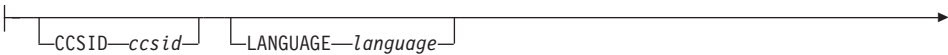
命令语法



attribute list:

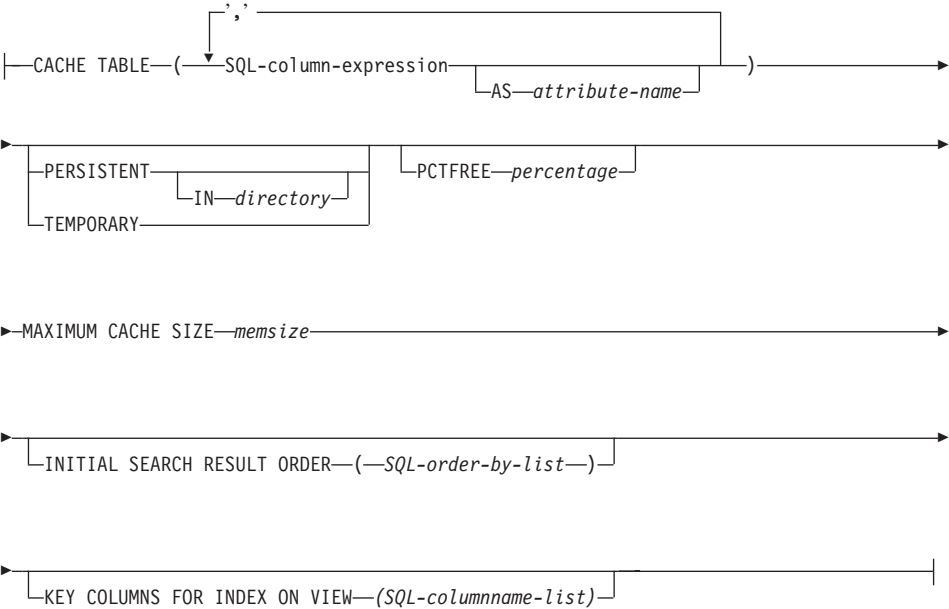


text-default-information:

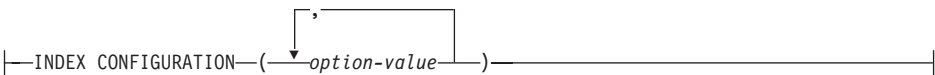


CREATE INDEX 命令

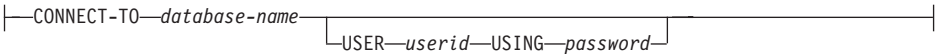
cache-search-results-options:



index-configuration-options:



connection-options:



命令参数

index schema

文本索引的模式。将该模式用作特定于索引的管理表的 DB2 模式名。如果没有指定任何模式，则使用 DB2 连接的用户标识。注意，索引模式必须是有效的 DB2 模式名。

index name

索引的名称。它与索引模式一起用来唯一地标识数据库中的全文本索引。它还用作索引事件表的名称。

有关详细信息，参见第 211 页的附录 C，『Net Search Extender 信息目录』。注意，索引名必须是有效的 DB2 索引名。

table schema

为其创建索引的表、昵称或视图的模式。如果没有指定任何模式，则使用 DB2 连接的用户标识。

table name

在已连接的数据库中包含为其创建了全文本索引的列的文本表、昵称或视图的名称。

注意，当表名不是指 DB2 基本表时，具有下列限制：

- 视图只允许进行存储过程或表值函数搜索。因此，必须使用 **KEY COLUMNS FOR INDEX ON VIEW** 子句指定索引或视图的键列。
- 对于对不具有捕获表的昵称的增量索引更新，会创建一个日志表。如果对昵称表或视图中的数据进行任何更改，则必须手工填写此信息。对于基本表，此操作是自动完成的，因此用户**一定不能**修改日志表。有关日志表的布局，参见第 211 页的附录 C，『Net Search Extender 信息目录』。
- 只有基本表或昵称而不是视图的索引才允许 DB2 谓词 **CONTAINS**、**SCORE** 和 **NUMBEROFMATCHES**。
- 仅当您在命令中指定 **cache-search-result** 选项时才允许视图的索引。

text-column-name

包含用于创建全文本索引的文本的列名。一列必须具有下列类型之一：

- **CHAR**（对于位数据）
- **VARCHAR**（对于位数据）
- **LONG VARCHAR**（对于位数据）
- **CLOB**
- **DBCLOB**
- **BLOB**
- **GRAPHIC**
- **VARGRAPHIC**
- **LONG VARGRAPHIC**
- **DATALINK**

如果列类型不属于以上任何类型，则使用

function-schema.function-name 指定变换函数用来转换列类型。

注意，如果使用 Data Link 列，则**一定会**取装引用的内容用来建立索引。这是通过作为 Data Link 值的一部分的协议（例如，Http）来实现的。当使用除了“file”或“unc”之外的协议时，确保服务器支持作为 Data Link

值的一部分的这些协议。由于获取文件内容可能需要代理服务器，因此，数据库管理员可以在创建索引之前在 DB2EXT.PROXYINFORMATION 表中指定它们。

注意，允许具有相同列的几个索引，但是，只有满足下列条件时才允许：

索引是对视图创建的

因此，不能在 CONTAINS、SCORE 或 NUMBER OF DOCUMENTS 搜索自变量中使用索引。

索引是对表创建的

如果所有索引都是同步的，则它们在下列 CREATE INDEX 命令详细信息中同一个列上具有完全相同的属性：

- Function name and schema
- ATTRIBUTES
- CCSID
- LANGUAGE
- FORMAT
- DOCUMENTMODEL
- INDEX CONFIGURATION

因此，CONTAINS、SCORE 或 NUMBER OF DOCUMENTS 自变量选择哪个索引并不重要。

function-schema.function-name

用来存取在具有不受支持类型的一列中的文本文档的用户定义函数的模式和名称。该函数使用任意列类型的一个输入参数来执行列类型转换。它返回 Net Search Extender 支持的其中一种类型的值。

ATTRIBUTES (SQL-column-expression AS Attribute-name, ...)

确保除了文本列之外还对列表表达式的内容建立了索引。也可以通过搜索语句中的 ATTRIBUTE 子句来搜索此内容。必须使用对其创建索引的表的非限定列名来定义 SQL 列表表达式。唯一允许的数据类型为双精度型。可以在列表表达式中使用强制转型运算符，但是，不可能进行 DB2 的隐式强制转型。属性名必须遵循文档模型中的属性名的规则，并且必须与索引模型定义文件中的属性名不同。

通过使用下列规则来确定表达式的属性名：

- 如果在列表表达式中由 SQL AS 子句显式命名，则使用指定的名称。一个示例是：ATTRIBUTES (C1+C2 AS myname)
- 如果不带 AS 使用指定表的列，则使用该列的名称。例如：CACHE TABLE (C1)

- 如果不带 AS 使用表达式，并且表达式不引用已命名的列，则 CREATE INDEX 会报告错误。

例如: ATTRIBUTES (CAST(JULIAN_DAY(date) AS DOUBLE) as day, (price1+price2)/2 as avg_price)

注意：不带引号的属性将映射为大写，并且必须在搜索期间以此方式指定。

CCSID ccSID

当对文本文档建立索引时，使用“编码字符集标识符”。缺省值来自 DB2EXT.DBDEFAULTS 视图，其中 DEFAULTNAME='CCSID'。

LANGUAGE language

有关列表，参见第 229 页的附录 E，『受支持的语言』。缺省值来自 DB2EXT.DBDEFAULTS 视图，其中 DEFAULTNAME='LANGUAGE'。

FORMAT format

一列中的文本文档的格式，例如，HTML。此信息对于为文档建立索引是必需的。有关结构化文档支持的文档格式的列表，参见第 25 页的『文档格式和受支持的代码页』。

对于结构化文档格式，可以在文档模型文件中指定信息。如果没有指定任何文档模型，则使用缺省文档模型来对文档的文本建立索引。参见第 185 页的『文档模型』。

如果没有指定格式关键字，则缺省值来自 DB2EXT.DBDEFAULTS 视图，其中 DEFAULTNAME='FORMAT'。

DOCUMENTMODEL documentmodel-name IN modelfilepath

modelfilepath 指定模型文件的位置。这包含 FORMAT 子句中的格式的模型定义。DB2 实例所有者必须可以读取它。文档模型使您能够对文档的特定部分建立索引和进行搜索。可以在文档模型中定义标记（markup tag）和节名。将文档模型绑定至支持 HTML、XML 或 GPP 结构的文档格式。在一个模型文件中只能指定一种文档模型。

由于在搜索条件中不需要引用文档模型，因此，使用模型文件中的所有节名。有关文档模型的详细信息，参见第 97 页的第 9 章，『使用结构化文档』。注意，由于只有在执行 CREATE INDEX 命令期间才读取文档模型，因此，不能识别此索引的任何后续更改。

注意，在分布式 DB2 环境中，在每个节点上都必须能够使用共享文件系统来存取 modelfilepath。

CREATE INDEX 命令

USING CCSID `ccsid`

指定 CCSID 来解释模型文件的内容。缺省值来自 DB2EXT.DBDEFAULTS 视图，其中 DEFAULTNAME='MODELCCSID'。

UPDATE FREQUENCY

索引更新频率确定何时进行更新。如果对用户表的更改数小于由 UPDATE MINIMUM 选项指定的更改数，则不会更新索引。如果不指定 UPDATE FREQUENCY，则使用缺省值 NONE，以便不再进行任何索引更新。这在不将不对文本列进行任何更改时很有用。

- **D.** 在星期几更新索引: * (每天) 或 0.6 (0 = 星期天)
- **H.** 在几点钟更新索引: * (每小时) 或 0.23
- **M.** 在多少分钟更新索引: 0.59
- **NONE.** 不再执行进一步的索引更新。必须手工启动更新。

缺省值来自 DB2EXT.DBDEFAULTS 视图，其中 DEFAULTNAME='UPDATEFREQUENCY'。

UPDATE MINIMUM `minchanges`

在 UPDATE FREQUENCY 自动更新索引之前允许对文本文档进行的最小更改数。允许正整数值。缺省值取自 DB2EXT.DBDEFAULTS 视图，其中 DEFAULTNAME='UPDATEMINIMUM'。

注意，在 DB2TEXT UPDATE 命令中会忽略此值。此选项不能与 RECREATE INDEX ON UPDATE 选项配合使用，原因是没有日志表和增量更新的触发器，更改数不可用。

对于分布式数据库，对每个节点检查 UPDATE MINIMUM。

REORGANIZE AUTOMATIC/MANUAL

如果指定了 REORGANIZE AUTOMATIC，则使用更新频率执行的更新将只识别索引。此步骤是在更新之后根据 select REORGSUGGESTED from DB2EXT.TEXTINDEXES 的值自动完成的。

只能使用手工 UPDATE 命令并通过使用 REORGANIZE 选项来执行 REORGANIZE MANUAL。

如果省略 REORGANIZE 子句，则缺省值取自 DB2EXT.DBDEFAULTS 视图，其中 DEFAULTNAME='AUTOMATICREORG'。

有关 REORGANIZE 选项的进一步信息，参见第 151 页的『UPDATE INDEX』。

REPLICATION CAPTURE TABLE capture-table-schema.capture-table-name CONTROL TABLE SCHEMA capture-control-schema

对于增量更新处理，采用指定的复制捕获表而不是通常为索引创建的日志表。因此，`schemaname`、`tablename` 和复制捕获表名与本地 DB2（联合）数据库中的对象相关。

`capture-control-schema` 是复制控制表的模式名，例如，本地 DB2 上的 `IBMSNAP_PRUNE_SET`。在设置了复制之后，复制控制表必须可用作本地 DB2 系统上的昵称。

至少，必须具有可用于下列捕获控制表的昵称：

- `IBMSNAP_SIGNAL`
- `IBMSNAP_PRUNE_SET`
- `IBMSNAP_PRUNCNTL`
- `IBMSNAP_REGISTER`
- `IBMSNAP_REG_SYNC`（仅适用于非 DB2 远程源）

由于“DB2 复制中心”不会自动保证为远程捕获表和捕获控制表创建昵称，这是类似于为要对其创建文本索引的表创建昵称的手工任务。

用户表昵称与捕获表昵称中的主键列的列名必须匹配。另外，一定不能更改捕获表昵称中的 `IBMSNAP_OPERATION`、`IBMSNAP_COMMITSEQ` 和 `IBMSNAP_INTENTSEQ` 列的名称。

创建索引之后，列名 `DB2EXT.TEXTINDEXES(LOGVIEWNAME)` 和 `DB2EXT.TEXTINDEXES(LOGVIEWSCHEMA)` 都表示复制捕获表的本地名称。

由于 Net Search Extender 并不需要“DB2 复制中心”的所有功能，所以“更改数据”（CD）表或一致更改数据（CCD）表必须遵守下列规则：

- 使用更改捕获注册但不使用完全刷新复制选项。
- 不允许捕获更改的任何水平子集。例如，通过触发器。参见 *DB2 Replication Guide and Reference, Version 8* 的第六章“Subsetting data in your replication environment”。
- 仅当包括主键列、文本列和 `DB2TEXT CREATE INDEX` 命令的属性和高速缓存表表达式中涉及到的所有列时，才允许注册列子集的更改。
- 主键列必须包括在捕获表中。注意，后映象是足够的。
- 一定不能压缩捕获表。对于每个主键，必须具有一个条目具有最新数据。但是，DB2 Net Search Extender 要求所有历史都可用。

CREATE INDEX 命令

- 表必须使用 D/I 选项。这使得能够将对源表上的主键的更新变换成插入 / 删除对。

其它先决条件包括:

- 对其创建索引的源表的服务器类型和版本是下列各项之一:
 - DB2/AIX V8.1 或更新版本
 - DB2/NT V8.1 或更新版本
 - DB2/HP V8.1 或更新版本
 - DB2/LINUX V8.1 或更新版本
 - DB2/SUN V8.1 或更新版本
 - DB2 z/OS® V8.1 或更新版本
 - DB2 OS/400 V5.2 或更新版本
 - Informix® IDS 9.3
 - ORACLE 9i
 - SYBASE ASE 12.5
 - Microsoft SQL Server 2000
- 受支持的包装器包括:
 - DB2: DRDA®
 - Informix: Informix
 - ORACLE: NET8 和 (SQLNET)
 - SYBASE: CTLIB
 - MSSQLSERVER: MSSQLODBC3

注意事项和限制

确保将正确的源表名插入到注册表中。视远程 DBMS 的类型不同，必须使用远程表名或本地昵称：

- DB2: 远程表名（远程服务器上的表名）
- 非 DB2: 本地昵称（联合 DB2 数据库中的相应昵称）

必须存在用户映射本地用户才能通过昵称来存取远程数据源，而远程用户必须对表具有 **CONTROL** 特权。

如果 DB2 实例所有者用户标识不同于本地用户标识，则需要实例所有者用户标识的附加用户映射。

指定的基本表名一定不能是昵称的视图。这是因为视图可建立在若干个昵称基础之上，并且还可能涉及到若干个 CD 表和 CCD 表。由于在复制捕获子句中只能指定一个 CD 表或 CCD 表，所以不能支持昵称的视图。另外，不能支持远程视图的昵称，这是因为缺少主键。

C(C)D 表必须是昵称而不能是视图或别名。

有关 *DB2 Replication Guide and Reference Version 8* 的信息，参见第 viii 页的『相关信息』。

COMMITCOUNT FOR UPDATE count

对于增量更新处理，可以指定落实计数，有关进一步的信息，参见第 151 页的『UPDATE INDEX』。如果未指定它，则缺省值取自 DB2EXT.DBDEFAULTS 视图，其中 DEFAULTNAME='COMMITCOUNT'。

可以在 DB2EXT.TEXTINDEXES.COMMITCOUNT 中找到索引的 COMMITCOUNT FOR UPDATE 值。可以使用 ALTER INDEX 命令对每个索引更改此值。它还适用于根据 UPDATE FREQUENCY 规范的已调度更新处理。如果值为 0，则意味着更新是在一个事务中完成的，如果值大于 0，则指定要在一个事务中处理的文档数。

使用 commitcount 会对性能有影响。有关信息，参见第 50 页的『性能注意事项』。

RECREATE INDEX ON UPDATE

这不允许增量索引更新，但是会在（由命令或已调度的更新）执行更新操作时重新创建索引。有关附加信息，参见第 151 页的『UPDATE INDEX』上的“使用说明”。

注意

没有对用户表创建任何触发器，也没有创建日志表。

INDEX DIRECTORY directory

要用来存储文本索引的目录路径。由于该目录将包含索引数据，因此应确保 DB2 实例所有者用户标识对该目录具有读 / 写和执行许可权。

缺省值取自 `DB2EXT.DBDEFAULTS` 视图，其中 `DEFAULTNAME=INDEXDIRECTORY`。在该目录下创建了一个子目录 `NODE<nr>` 以区分服务器的逻辑节点上的索引。

注意，在分布式 DB2 环境中，每个物理节点上都必须存在此目录。

WORK DIRECTORY directory

（可选）可以指定独立的工作目录，该目录将在执行索引搜索和管理操作期间用来存储临时文件。该目录必须存在，并且对于 DB2 实例所有者标识都具有读 / 写和执行许可权。

缺省值取自 `DB2EXT.DBDEFAULTS` 视图，其中 `DEFAULTNAME='WORKDIRECTORY'`。在该目录下创建了一个子目录 `NODE<nr>` 以区分服务器的逻辑节点上的索引。

注意，在分布式 DB2 环境中，每个物理节点上都必须存在此目录。

ADMINISTRATION TABLES IN tablespace-name

为索引创建的管理表的常规表空间的名称。表空间必须存在。如果不指定此项，则会选择用户表的表空间（如果索引是对基本表创建的）。

对于昵称或视图，DB2 将选择缺省表空间。

当为分布式 DB2 环境中的存储过程搜索而对视图、昵称或文本索引创建文本索引时，表空间必须属于单个节点。

CACHE TABLE (SQL-column-expression-list)

除了索引之外，还会构建高速缓存表，该表由指定的列表表达式组成。此高速缓存用来通过存储过程搜索返回结果集而不必将全文本搜索结果与 DB2 表连接起来。注意，始终可以执行将全文本索引与 `CONTAINS` 函数配合使用的常规 DB2 搜索。

使用对其创建索引的表的非限定列名来定义 SQL 列表表达式。允许的 SQL 列表表达式类型是所有内置的和用户定义的单值类型。结果集中的列名是使用下列规则确定的：

- 如果在列表表达式中由 SQL AS 子句显式命名，则使用指定的名称。例如：CACHE TABLE (C1+C2 AS myname)
- 如果不带 AS 子句使用指定表的列，则使用该列的名称。例如：CACHE TABLE(C1)
- 如果不带 AS 使用表达式，并且表达式不引用已命名的列，则 CREATE INDEX 会报告错误。
- 不允许重复的列名。

不支持将 CLOB 数据类型作为高速缓存数据类型。需要将这些数据类型强制转型为 VARCHAR。

注意

注意，如果结果集的列名不是互不相同的，则 CREATE INDEX 命令会返回错误。还要注意，在创建之后，高速缓存表不是隐式激活的，例如，在执行 DB2TEXT ACTIVATE CACHE 之前，不可能执行由存储过程执行的搜索。

仅当用户表存储在具有单个节点的表空间中时，才能在分布式 DB2 环境中使用此选项。

PERSISTENT IN directory

指定创建的高速缓存是持久的，并且可在取消激活或系统重新引导之后短暂激活。持久高速缓存存储在指定的目录中。

注意，如果没有指定目录，则缺省值取自 DB2EXT.DBDEFAULTS 视图，其中 DEFAULTNAME='CACHEDIRECTORY'。

TEMPORARY

指定高速缓存不是持久存储的。如果既没有指定 PERSISTENT，也没有指定 TEMPORARY，则缺省值取自 DB2EXT.DBDEFAULTS 视图，其中 DEFAULTNAME='USEPERSISTENTCACHE'。

MAXIMUM CACHE SIZE memsize

指定在 DB2TEXT ACTIVATE CACHE 期间要构建的高速缓存表的最大大小。必须以兆字节为单位将 memsize 参数指定为正整数。没有为 memsize 指定缺省值。如果该整数太小，则 ACTIVATE CACHE 命令将失败。实际高速缓存大小是在执行 ACTIVATE CACHE 命令期间计算的。

不同平台上对最大高速缓存大小的限制是:

- Windows: 1024 MB (1 GB = 1073741824 个字节)
- AIX: 1536 MB (1.5 GB = 1610612736 个字节)
- Solaris、Linux 和 HP-UX: 2048 MB (2 GB = 2147483647 个字节)

有关更多信息, 参见第 207 页的附录 B, 『使用大量内存』。

PCTFREE percentage

指定要为附加文档保留的高速缓存的百分比。该百分比必须是小于 100 并且大于或等于 0 的整数。如果未指定此项, 则缺省值取自 DB2EXT.DBDEFAULTS 视图, 其中 DEFAULTNAME='PCTFREE'。

有关详细信息, 参见第 122 页的『ACTIVATE CACHE』。

INITIAL SEARCH RESULT ORDER (SQL-order-by-list)

指定在初次建立索引期间用于检索用户表内容的次序。当使用此选项时, 通过跳过对全文搜索结果进行动态排序, 文档就会象存储在高速缓存结果表中一样按它们的建立索引次序返回。

有关进一步的信息, 参见第 179 页的第 16 章, 『存储过程搜索函数』。

注意

对于增量更新之后的新文档或已更改的文档, **不能**确保该索引次序。例如: INITIAL RESULT ORDER(length(column1) asc, column2+column3 desc)

KEY COLUMNS FOR INDEX ON VIEW (SQL-columnname-list)

如果对视图创建了索引, 则必须指定 KEY COLUMNS FOR INDEX ON VIEW 子句, 否则, “一定不能”指定该子句。列名列表指定“唯一地”标识视图中的一行的列。

由于 DB2 不能对主键检查这种唯一性, 因此, 用户应负责确保等价的唯一性。指定的列为索引构建日志表的一部分。

INDEX CONFIGURATION (option-value), ...

这些是索引配置值。缺省值加上了下划线。

选项	值	描述
TreatNumbersAsWords	<u>0</u> 或 1	将一系列数字解释为独立的词语, 即使它们与字符相邻也是这样, 例如, 缺省值 0 意味着将 tea42at5 看作是一个词语。

IndexStopWords	0 或 1	在建立索引期间考虑或者忽略无用词。目前，无用词列表是 <instance>/sql1lib/db2ext/resources 目录中的一个 UCS-2 文件 <language>.tsw。创建索引之后，更改此文件没有任何影响。还要注意，<language> 是 CREATE INDEX 命令中的 LANGUAGE 值。
UpdateDelay	秒	指定不使用捕获表的增量更新的持续时间（以秒计）。将仅从日志表中抽取早于此持续时间的条目。这将避免丢失更新，例如，在用户事务对更新命令有影响的事务情况中，未反映在索引中的文档更改。因此，UpdateDelay 参数应设置为对其创建索引的表的用户写事务的最长持续时间。

CONNECT TO database-name

作为此命令的目标的数据库的名称。如果设置了 DB2DBDFT 并且用户正在服务器上运行命令，则可以省略此参数。注意，用户标识必须具有必需的 DB2 权限。

USER userid USING password

使用 password 和 userid 来连接至数据库。如果未指定它们，则会尝试使用当前用户标识而不带密码来建立连接。

对数据库的更改

- 更改 DB2 Net Search Extender 目录视图。
- 在指定的表空间中创建索引日志表。仅当未指定 RECREATE INDEX 选项并且未指定捕获表时才执行此操作。
- 在指定的表空间中创建索引事件表。
- 因循第一次更新：对用户文本表创建触发器（仅当未指定 RECREATE INDEX 并且未使用捕获表时）
- 如果使用了复制捕获表，则会对捕获控制表进行下列更改：
 - 插入到 IBMSNAP_PRUNCTNL 和 IBMSNAP_PRUNE_SET 表中这些表中的条目是由下面的列唯一标识的：
 - APPLY_QUAL='NSE' || <DB2 instance running NSE>
 - SET_NAME= <internal index identifier>
 - TARGET_SERVER=<DB2 database name target to DB2TEXT operation>

有关列的更多信息，参见 39。

对共享内存的更改

因循 ACTIVATE 执行：如果使用了 CACHE TABLE 子句，则会在共享内存中构建结果表的高速缓存。

对文件系统的更改

- 在索引目录、工作目录和高速缓存目录下创建了 NODE<nr> 子目录。
- <internal index name> 目录是在 <indexdirectory>/NODE<nr> 下创建的，其中 indexdirectory 指的是此命令的相应参数，NODE<nr> 与分布式 DB2 环境中的节点号相关。

使用法

创建全文本索引需要用户表的主键。在 DB2 Net Search Extender V8.1 中，可以使用包含多列的 DB2 主键，没有任何类型限制。但是，要使用表值搜索，不允许任何复合主键。

主键列的数目最多为 14 列，所有主键列的总长度最长为 $1024 - 14 = 1010$ 个字节。

- ATTRIBUTES、CACHE TABLE 和 INITIAL SEARCH RESULT ORDER 的 SQL 表达式的总大小一定不能超过 24K 字节。
- 初始索引更新总是作为一个逻辑事务来完成的，在这种情况下不存在落实计数。

注意

在创建索引之后，一定不能使用 ALTER TABLE 命令来更改主键列或视图键列的长度。

用户表、全文本索引和高速缓存的结果表之间的同步是在执行更新索引命令期间完成的。有关进一步的信息，参见第 151 页的『UPDATE INDEX』。

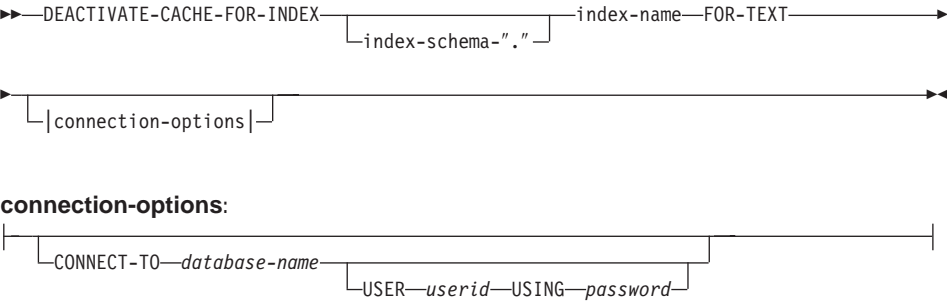
DEACTIVATE CACHE

此命令将释放高速缓存的表。保留持久高速缓存以便在下一个 ACTIVATE 命令中重新使用。在下次激活之前，不再可以在取消激活的高速缓存中执行通过存储过程执行的搜索操作。

授权

根据 DB2 目录视图，此命令中的用户标识必须对为其创建全文索引的表具有 CONTROL 特权。

命令语法



命令参数

index-schema

文本索引的模式，如在 CREATE INDEX 命令中指定的那样。如果没有指定任何模式，则使用 DB2 连接的用户标识作为模式名。

index-name

文本索引的名称，如在 CREATE INDEX 命令中指定的那样。

CONNECT TO database-name

作为此命令的目标的数据库的名称。如果设置了 DB2DBDFT 并且用户正在服务器上运行命令，则可以省略此参数。注意，用户标识必须具有必需的 DB2 权限。

USER userid USING password

使用 password 和 userid 来连接至数据库。如果未指定它们，则会尝试使用当前用户标识而不带密码来建立连接。

使用法

注意，当正在对索引运行下列命令之一时，不能发出此命令：

- ACTIVATE CACHE

DEACTIVATE CACHE 命令

- DEACTIVATE CACHE
- UPDATE INDEX
- ALTER INDEX
- DROP INDEX
- CLEAR EVENTS

注意

在取消激活持久高速缓存之后，通过存储过程执行的搜索就不能存取该高速缓存。但是，除非同时执行更新，否则，这可用于快速 **ACTIVATE**。

在这种情况下，将使用 **ACTIVATE CACHE** 命令自动从头开始重新创建持久高速缓存。

DROP INDEX

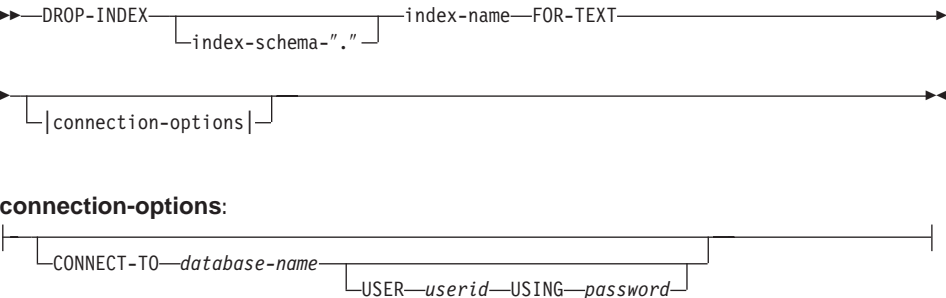
此命令将删除文本列的全文本索引。如果激活了索引的高速缓存，则使用此命令来删除索引。

授权

根据 DB2 目录视图，此命令中的用户标识必须对为其创建全文本索引的表具有 CONTROL 特权。或者，用户可以是数据库管理员（DBADM）。

或者，数据库管理员（DBADM）可以删除索引，原因是他们必须能够使用 FORCE 选项来禁用数据库。

命令语法



命令参数

index schema

文本索引的模式，如在 CREATE INDEX 命令中指定的那样。如果没有指定任何模式，则使用 DB2 连接的用户标识作为模式名。

index-name

索引的名称，如在 CREATE INDEX 命令中指定的那样。借助索引模式，索引名将唯一地标识数据库中的全文本索引。

CONNECT TO database-name

作为此命令的目标的数据库的名称。如果设置了 DB2DBDFT 并且用户正在服务器上运行命令，则可以省略此参数。注意，用户标识必须具有必需的 DB2 权限。

USER userid USING password

使用 password 和 userid 来连接至数据库。如果未指定它们，则会尝试使用当前用户标识而不带密码来建立连接。

使用法

删除索引，而不管其高速缓存表的激活状态如何。有关附加信息，参见第 122 页的『ACTIVATE CACHE』以获得更多信息。

注意，当正在对索引运行下列命令之一时，不能发出该命令：

- UPDATE INDEX
- CLEAR EVENTS
- ALTER INDEX
- ACTIVATE CACHE
- DEACTIVATE CACHE
- DROP INDEX

注意

在删除 DB2 中的用户表前后，必须手工删除索引。如果不手工删除索引，则不能正确清理结果。

对数据库的更改

- 更改 DB2 Net Search Extender 目录视图
- 删除 DB2 索引
- 删除索引日志表 / 事件表
- 删除用户文本表的触发器

当使用复制捕获表时，会除去 IBMSNAP_PRUNE_SET 和 IBMSNAP_PRUNCTRNL 表中的条目。

对共享内存的更改

删除了高速缓存的表。

对文件系统的更改

- 删除了已删除索引的索引目录和工作目录中的 <internal index name> 目录
- 删除索引的持久高速缓存

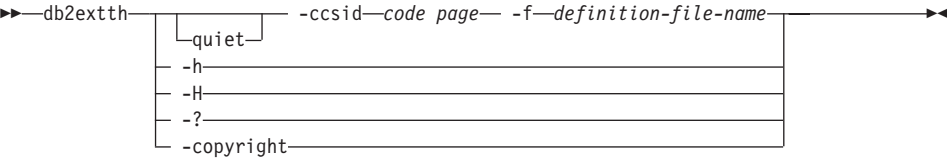
DB2EXTTH (实用程序)

这个独立的实用程序将编译同义字定义文件。运行了同义字编译器之后，就可以使用搜索自变量语法的与 THESAURUS 相关的功能部件。

授权

无。不需要对表所有者限制此命令，但是只在查询的上下文中有意义。

命令语法



命令参数

-f definition-file-name

包含同义字定义的文件名称。该文件名必须包含文件的绝对路径或相对路径。该文件名最多为 8+3 个字符，扩展名是可选的。

同义字典是在定义文件所在的目录中生成的，并且它们名称相同。唯一的区别是字典具有下列扩展名：

wdf、wdv、grf、grv、MEY、ROS、NEY、SOS 和 lkn，其中 n 是一个数字。注意，如果现有同义字文件具有相同名称，则会覆盖它们。

-ccsid code page

编写同义字定义文件所使用的代码页。有关同义字的受支持的代码页列表，参见第 275 页的附录 L，『同义字支持的 CCSID』。

-quiet 不显示输出信息。

-copyright

返回产品的内部构建号。当报告问题时使用此编号。

-h、-H 或 -?

显示帮助信息。

使用法

使用此命令来将同义字定义文件编译为二进制同义字定义格式。

注意

格式与 DB2 Text Information Extender V7.2 相同，但是进行了下列更改：

- 新关系 **BROADER** 和 **NARROWER** 等价于先前在 Text Information Extender 中使用的 **HIGHER_THAN** 和 **LOWER_THAN** 关系。要引用这些旧的搜索关系，必须使用新关系。
- 还要注意，同义字典文件必须存储在 `<os-dependent>/sql1lib/db2ext/thes` 中，以便在搜索期间可以使用，除非同义字在查询中是全限定的。

有关附加信息，参见第 99 页的第 10 章，『使用同义字来扩展搜索术语』。

UPDATE INDEX

此命令通过显示最新索引来立即启动建立索引过程，以反映与索引相关联的文本列的当前内容。

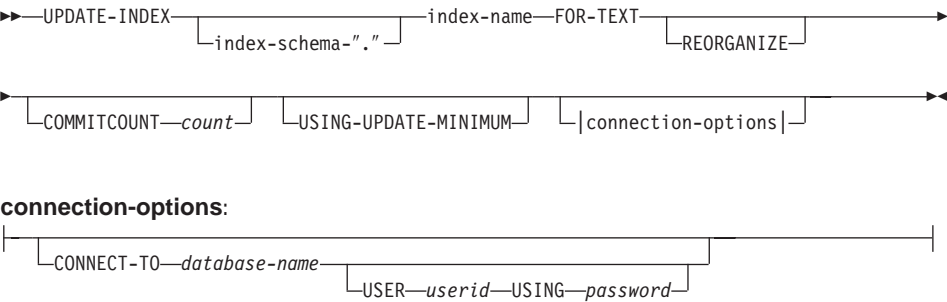
当正在执行更新时，可以执行使用 CONTAINS 谓词的搜索。对于具有已激活的高速缓存结果表的索引，在更新期间，也可以执行通过存储过程执行的搜索。但是，高速缓存表中的列可能会显示新值，即使尚未将已更改的文本落实到全文本索引中也是如此。

在 CREATE INDEX 命令中使用 RECREATE INDEX ON UPDATE 选项将在重新创建之前清除索引。在完成更新之前，将返回空结果。

授权

根据 DB2 目录视图，此命令中的用户标识必须对为其创建全文本索引的表必须具有 CONTROL 特权。

命令语法



命令参数

index-schema

文本索引的模式。这是在 CREATE INDEX 命令中指定的。如果没有指定任何模式，则使用 DB2 连接的用户标识。

index-name

文本索引的名称。这是在 CREATE INDEX 命令中指定的。

REORGANIZE

如果频繁更新文本列，则对索引的后续更新可能会变得不足。要使更新过程再次变得是足够的，可以重组索引。使用 DB2EXT.TEXTINDEXES 视图来确定是否需要重组索引。

UPDATE INDEX 命令

使用 CREATE INDEX 命令的 REORGANIZE AUTOMATIC 选项来避免手工检查和重组索引。

注意

重组过程是在常规更新之后进行的。

USING UPDATE MINIMUM

仅当达到了指定的更改数时才在 CREATE INDEX 命令中使用 UPDATE MINIMUM 设置并启动增量更新。缺省值是无条件启动更新。

对于分布式数据库，对每个节点检查 UPDATE MINIMUM。

有关附加信息，参见第 130 页的『CREATE INDEX』。

COMMITCOUNT count

≥ 0 的 INTEGER 值显示由搜索引擎和由 DB2 在一个事务中为增量索引更新处理的文档数。

但是，对于首次更新，例如，执行 CREATE INDEX 命令之后的第一次更新或者使用 RECREATE INDEX ON UPDATE 选项进行的任何更新，只存在在一个逻辑事务，它忽略了 COMMITCOUNT。可以使用 ALTER INDEX 命令来更改它。

CONNECT TO database-name

作为此命令的目标的数据库的名称。如果设置了 DB2DBDFT 并且用户正在服务器上运行命令，则可以省略此参数。注意，用户标识必须具有必需的 DB2 权限。

USER userid USING password

使用 password 和 userid 来连接至数据库。如果未指定它们，则会尝试使用当前用户标识而不带密码来建立连接。

使用法

此命令以同步方式运行。它在分布式 DB2 环境中的所有必需的 DB2 逻辑 / 物理节点上启动更新处理。持续时间取决于要建立索引的文件数和已经建立索引的文档数。可以通过为每个索引创建的视图来查看更新的状态。可以从 DB2EXT.TEXTINDEXES 的 EVENTVIEWNAME 列中检索此视图的名称。有关进一步的信息，参考第 211 页的附录 C，『Net Search Extender 信息目录』。

可以使用两个选项来查看已经处理的已落实文档的数目。要确定一个更新是否仍然在运行以及已经将多少个文档落实到索引中，使用

DB2EXT.TEXTINDEXES (NUMBERDOCS) 视图。使用与索引相关联的事件视图以获取有关启动、落实更改和完成更新处理的信息。

要查看要处理的未落实的文档数，使用 CONTROL LIST ALL LOCKS FOR INDEX 命令。

注意

这些视图只显示来自于已连接的节点的信息。

对于具有物理节点的基本表的增量更新，每个节点上的时间必须同步。如果时间不同步，则更新可能丢失或根本不发生。

如果正在对索引运行下列命令之一，则不能发出该命令：

- CLEAR EVENTS
- ALTER INDEX
- DROP INDEX
- ACTIVATE CACHE
- DEACTIVATE CACHE
- UPDATE INDEX

在使用已取消激活的持久高速缓存结果表更新了索引之后，就会删除持久高速缓存，以便下一个 ACTIVATE CACHE 命令根据数据库内容重新创建它。

如果用户中断此命令，则更新功能涉及到的所有进程都会停止。如果在增量更新中使用了落实计数，则在索引中可能可以看见某些更新，而其它更新可能需要新的更新命令。

要停止自动更新索引，在用于更新服务的分区上查找正在运行更新索引命令的 DB2 实例所有者进程。停止所有分区上的此进程和更新处理。

注意

由于该命令在所有分区上的索引创建和初始索引更新的两个独立阶段起作用，因此，发出 db2text drop index 命令来确保索引不是部分可用。如果不发出此命令，则可能由更新命令或更新频率选项触发的下一次更新可能会执行完全重新建立索引来确保一致状态。

对数据库的更改

UPDATE INDEX 命令

- 插入到事件表
- 从索引日志表中删除

当使用复制捕获表时，对数据库进行了下列更改。

- 在启动初始更新之前将信号添加至 `IBMSNAP_SIGNAL` 表
- 在增量更新之后更改 `IBMSNAP_PRUNE_SET` 的同步点

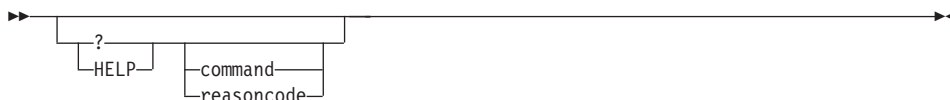
HELP

此命令显示可用的 DB2TEXT 命令的列表，或者显示个别 DB2TEXT 命令的语法。

授权

不需要任何权限。

命令语法



命令参数

HELP 或 ?

提供有关指定命令或原因代码的帮助。

command

用来标识 DB2TEXT 命令的第一个关键字：

- ENABLE
- DISABLE
- CREATE
- DROP
- ALTER
- UPDATE
- CLEAR
- START
- STOP
- CONTROL
- ACTIVATE
- DEACTIVATE

reasoncode

来自 DB2 Net Search Extender 命令的原因代码。

使用法

如果除了第一个关键字之外指定了多个关键字，则会忽略其余关键字，并且会显示所标识的命令的语法。

HELP 命令

如果在 “?” 或 “HELP” 后面没有指定 “command” 参数（或者根本不帶参数），则 DB2TEXT 将列示所有可用的 DB2TEXT 命令参数。

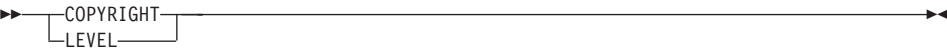
COPYRIGHT

提供 Net Search Extender 产品和版权信息。

授权

不需要任何权限。

命令语法



命令参数

COPYRIGHT / LEVEL

提供产品的版本版权语句、版本号和构建信息。

第 14 章 搜索自变量的语法

搜索自变量是您在文本文档中搜索术语时指定的条件。它由搜索参数以及一个或多个搜索术语组成。

在第 85 页的『指定 SQL 搜索自变量』和一个称为 `search` 的文件中提供了搜索自变量示例。参见第 90 页的『附加搜索语法示例』。

使用搜索自变量的 SQL 标量搜索函数包括：

CONTAINS

此函数使用搜索自变量来搜索特定文本文档中的文本。如果该文档包含文本或者在搜索自变量中指定的任何关系，则将返回 `INTEGER` 值 1。否则，它将返回 0。

NUMBEROFMATCHES

此函数使用搜索自变量来搜索文本文档，然后返回一个指示在每个文档中找到了多少个匹配项的 `INTEGER` 值。

SCORE

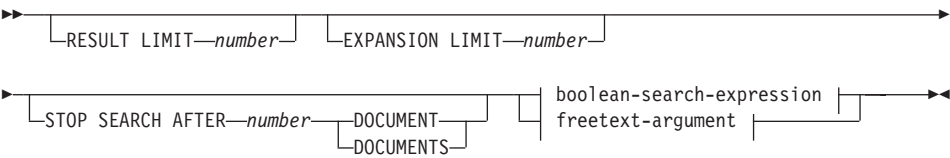
此函数使用搜索自变量来搜索文本文档。它为找到的每个文档返回一个值，指示搜索自变量是如何描述找到的文档的。

注意

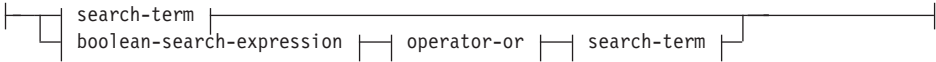
在存储过程搜索和 SQL 表值函数的搜索自变量中使用相同的语法。

搜索自变量

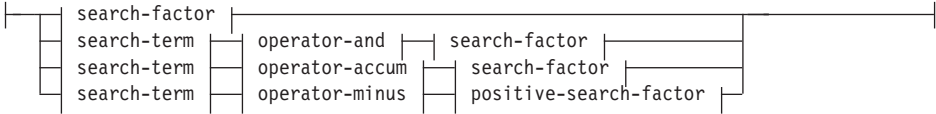
搜索自变量语法



Boolean-search-expression:



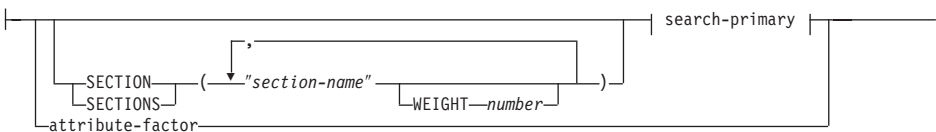
search-term:



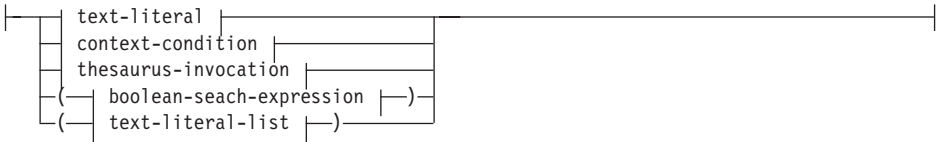
Search-factor:



Positive-search-factor:



Search-primary:



Operator-and:



Operator-or:



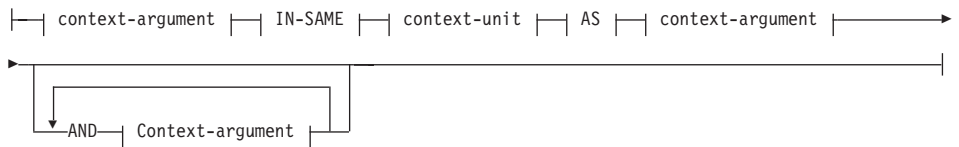
Operator-accum:



Operator-minus:



Context-condition:



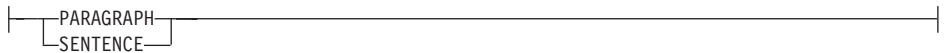
Context-argument:



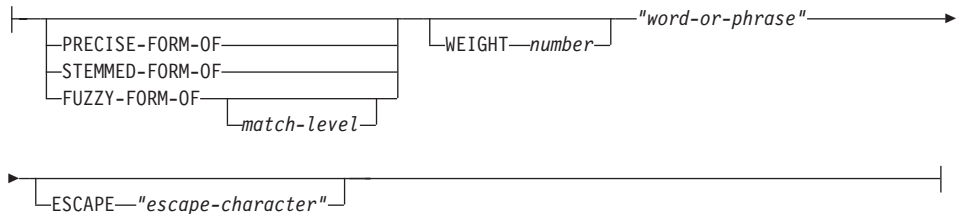
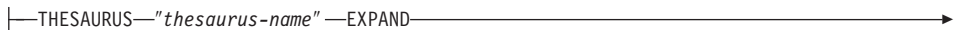
Text-literal-list:

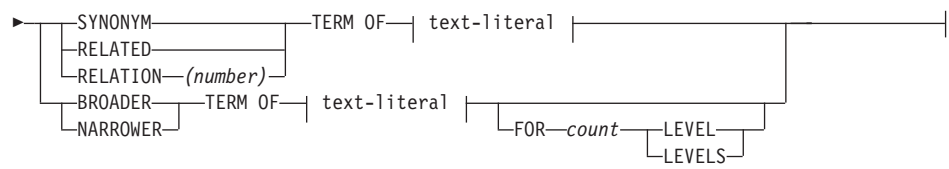


Context-unit:

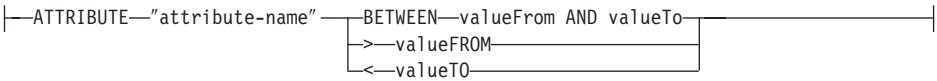


Text-literal:

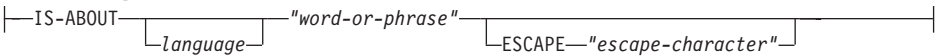
**thesaurus-invocation:**



Attribute-factor:



freetext-argument:



示例

在第 85 页的『指定 SQL 搜索自变量』中提供了示例。

搜索参数

RESULT LIMIT number

一个关键字，它指定要由全文本搜索返回的最大结果数。

RESULT LIMIT 应当与 SCORE 函数一起使用来确保返回的结果已计算了分数，并且只处理最佳结果。

EXPANSION LIMIT number

一个关键字，它指定可以为了进行搜索而扩展一个术语的最大次数。例如，确定可以扩展搜索术语“a*”的次数。

STOP SEARCH AFTER number DOCUMENTS(S)

用来指定搜索阈值的一个关键字。如果在搜索期间达到了文档数，则搜索将停止，并返回中间结果。较小的值可提高搜索性能，但是可能会导致获得较少的结果，并且会省略可能具有高级别的文档。

注意，此关键字没有缺省值，并且 number 值必须是正整数。

boolean-search-expression

可以根据语法图并使用布尔运算符 NOT、AND、OR、ACCUM 和 MINUS 来组合搜索术语和搜索因子。这些运算符具有以下优先权次序（第一个的优先权最高）：NOT> MINUS = ACCUM = AND > OR。可以在以下示例中了解到这种情况：

"Pilot" MINUS "1passenger" &"vehicle" | "transport" & "public"

它将按以下次序进行求值:

```
((("Pilot" MINUS "passenger") & ("vehicle")) | ("transport" & "public"))
```

如果其中一个布尔自变量求值为 `true` (可与 `OR` 运算符进行比较), 则 `ACCUM` 运算符求值为 `true`。等级值是通过累计两个操作数中的等级值来计算的。`ACCUM` 运算符与 `AND` 具有相同的绑定 (优先权)。如果左操作数求值为 `true`, 则 `MINUS` 运算符求值为 `true`。如果右边的操作数求值为 `true`, 则采用左边的操作数的等级值并减去损失来计算等级值。

search-primary

如果在 (文档的指定部分) 找到扩展的任何文本文字, 则由同义字调用组成的 `search-primary` 运算符求值为 `true`。如果在 (文档的指定部分) 中找到任何文本文字, 则由文本文字列表组成的 `search-primary` 求值为 `true`。

SECTION(S) *section-name*

一个关键字, 它指定要将搜索范围限制在结构化文档中的一个或多个部分。必须在创建索引时在指定的模型文件中指定这些部分的名称, 请参见第 130 页的『CREATE INDEX』。

这些部分的名称是区分大小写的。确保模型文件和查询中的部分名是完全相同的。

此模型描述包含可标识部分的文档的结构, 因此可以单独搜索这些部分的内容。不能使用屏蔽字符来屏蔽部分名。如果在某个部分找到 `search primary`, 则使用 `SECTION` 子句的 *positive-search-factor* 求值为 `true`。

context-argument IN SAME context-unit AS context-argument AND context-argument ...

此条件允许您在同一个段落或同一个句子中搜索文本文字的组合。上下文自变量始终等价于文本文字列表, 并且可以使用同义字扩展来将文本文字扩展为这样一个列表。

如果文档中具有上下文单元 (与段落有关的句子), 文档中的每个扩展的上下文自变量至少包含其中一个文本文字, 则此条件求值为 `true`。可以在以下示例中了解到这种情况:

```
("a","b") IN SAME PARAGRAPH AS ("c","d")
AND THESAURUS "t1" EXPAND SYNONYM TERM OF "e".
```

Assuming e1, e2 as synonyms of e, the following paragraphs would match:

```
".. a c e .." ,    ".. a c e1..",    "a c e2..",
".. a d e .." ,    ".. a d e1..",    "a d e2..",
".. b c e .." ,    ".. b c e1..",    "b c e2..",
".. b d e .." ,    ".. b d e1..",    "b d e2..".
```

PRECISE FORM OF

一个关键字，这使得要完全按照输入那样来搜索 PRECISE FORM OF 后面的词语（或者一个短语中的每个单词）。这种搜索是区分大小写的；即，到底使用大写字母还是小写字母是很重要的。例如，如果搜索 mouse，则找不到 “Mouse”。

STEMMED FORM OF

一个关键字，它在执行搜索之前将 STEMMED FORM OF 后面的词语（或者一个短语中的每个单词）缩短为词干。这种搜索不区分大小写。例如，如果搜索 mouse，则可以找到 “Mouse”。

采用哪种方式将词语缩短其词干形式是与语言有关的。目前，只支持英语，而且词语还必须遵循常规词尾变化。

FUZZY FORM OF

用于进行“模糊”搜索的一个关键字，“模糊”搜索就是搜索与搜索术语具有相似拼写的那些术语。当在由“光学字符识别”（OCR）程序创建的文档中进行搜索时，模糊搜索特别有用。这样的文档通常包括拼写错误的词语。例如，词语 economy 可能会被 OCR 程序识别为 econony。注意，前三个字符必须匹配，如果搜索源中的词语包含屏蔽字符，则不能使用模糊搜索。

match level

1 到 100 之间的一个整数，它指定相似程度，其中 100 比 1 更相似。100 指定“精确匹配”，60 被认为是一个非常“模糊值”。匹配级别越模糊，花费的搜索时间就越长，这是因为要搜索更多文档。缺省匹配级别为 70。

WEIGHT number

将文本文字与权值关联来更改缺省分数。允许的权值是 0（最低的分数量值）到 1000（最高的分数量值）之间的整数，缺省值为 100。

word-or-phrase

要搜索的词语或短语。在词语中可以使用的字符与语言有关。是否需要用分隔符将词语隔开，也是与语言有关的。对于英语和其它大多数语言，短语中的每个词语都必须用空格字符隔开。

要搜索包含双引号的字符串，应输入双引号两次。例如，要搜索文本 “wildcard” character，使用：

```
""wildcard"" character"
```

注意，在该示例中，只能搜索一组引号。不能够搜索一个序列中的两个引号。每个词语或短语的最大长度只能为 128 个字节。

Masking characters

一个词语可以包含下列屏蔽字符：

_ (下划线)

表示任何单个字符。

% (百分号)

表示任何数目的任意字符。如果一个词语由单个 % 组成，则它表示一个任意长度的可选词语。除了使用单个 % 来表示可选词语之外，一个词语不能只由屏蔽字符组成。如果使用了屏蔽字符，则不能使用“同义字”。屏蔽字符不能跟在非字母数字字符后面。

ESCAPE escape-character

一个字符，它将下一个字符标识为要搜索的字符，而不是要用作屏蔽字符。例如，如果转义字符是 \$，则 \$%、\$_ 和 \$\$ 分别表示 %、_ 和 \$。前面没有 \$ 的任何 % 和 _ 字符都表示屏蔽字符。

THESAURUS thesaurus-name

一个关键字，用来指定要用来扩展文本文字的同义字的名称。同义字名称是已经使用同义字编译器进行编译的同义字的文件名（不带扩展名）。它必须位于 <os-dependent>/sql1lib/db2ext/thes 中。或者，可以在文件名前面指定路径。

EXPAND relation

指定使用哪种关系来扩展使用同义字的文本文字。同义字已经预定义了 DB2EXTTH 命令中描述的关系。使用下列关键字来表示这些关系：

- SYNONYM，表示等价的对称关系。
- RELATED，表示关联的对称关系。
- BROADER，可以后跟指定的深度级别的直接分层关系。
- NARROWER，可以后跟指定的深度级别的直接分层关系。

对于用户定义的关系，使用 RELATION(number)，它对应于 DB2TEXTTH 中的关系定义。

TERM OF text-literal

要将其它搜索术语从同义字添加至的文本文字。

count LEVELS

一个关键字，用来指定同义字中术语的层数（深度），这些同义字将用来扩展给定关系的搜索术语。如果不指定此关键字，则假定 COUNT 为 1。深度的值必须是一个正整数值。

ATTRIBUTE Attribute-name

搜索具有与指定条件相匹配的属性的文档。属性名是指 CREATE INDEX 命令中的属性表达式的名称，或者是文档模型文件中的属性定义。

属性因子仅用于类型为 `double` 的属性。保证该值的精度为 15 位。16 个字符和 16 个字符以上的数字会被四舍五入。在 `attribute-name`、`valueFrom` 和 `valueTo` 中不允许使用屏蔽字符。有关解释，参见下列内容：

BETWEEN valueFrom AND valueTo

如果属性的值大于（不等于）`valueFrom` 并且小于（不等于）`valueTo`，则 **BETWEEN** 属性因子求值为 `true`。

>valueFrom

如果属性的值大于（不等于）`valueFrom`，则 “>” 属性因子求值为 `true`。

<valueTo

如果属性的值小于（不等于）`valueTo`，则 “<” 属性因子求值为 `true`。

如果 **CREATE INDEX** 命令中的属性名是使用引号来指定的，或者是在模型文件中定义的，则指定的属性名必须精确匹配。但是，如果 **CREATE INDEX** 命令中没有指定引号，则属性名必须采用大写。

IS ABOUT language word-or-phrase

一个允许您指定自由文本搜索自变量的选项。应当使用它来获取不同种类的分数的算法，原因是它会检查文档中的术语的定位。词语或短语中使用的术语互相越接近，文档中就包括更多术语，返回的分数值就越高。

第 229 页的附录 E，『受支持的语言』中描述了允许的语言值，它只与泰国语有关。如果未指定，则将 `en_US` 语言用作缺省语言。语言只用于单词或短语的标记。

注意，仅当请求了分数值，并且搜索结果按分数值排序时，**IS ABOUT** 才有用。

第 15 章 SQL 标量搜索函数和 SQL 表值函数

DB2 Net Search Extender 提供了 SQL 标量搜索函数和 SQL 表值函数来搜索存储在 “DB2 通用数据库” 中的文本文档。

本章描述了下列 SQL 搜索函数。

搜索函数总结

搜索函数	用途	页
CONTAINS	在特定文档中搜索文本。	168
NUMBEROFMATCHES	搜索并返回找到的匹配项数目。	169
SCORE	搜索并返回找到的文本文档的分数值。	170
DB2EXT.TEXTSEARCH	SQL 表值函数返回找到的主键、匹配项数目和 / 或分数值组成的表。	171
DB2EXT.HIGHLIGHT	获取有关将文档限定为搜索结果的原因的信息。	175

参见第 83 页的第 8 章, 『搜索』以获取使用 SQL 标量搜索函数和 SQL 表值函数的示例。

CONTAINS

CONTAINS 标量函数将搜索由 Net Search Extender 建立索引的文本文档中的文本。如果该文档包含文本或者在搜索自变量中指定的任何关系，则将返回 INTEGER 值 1。否则，它将返回 0。

函数语法

►►CONTAINS(—*column-name*—,—*search-argument*—)————►◄

函数参数

column name

列的名称。一列必须具有相关联的文本索引。可以使用管理命令 DB2TEXT CREATE INDEX 来创建文本索引。

search-argument

包含要搜索的术语的类型为 VARCHAR 的字符串。参见第 159 页的第 14 章，『搜索自变量的语法』。

注意

不能对为视图创建的文本索引使用 CONTAINS 查询。

NUMBEROFMATCHES

NUMBEROFMATCHES 标量函数搜索文本文档，然后返回一个指示在每个文档中找到了多少个匹配项的 INTEGER 值。

函数语法

►►—NUMBEROFMATCHES—(*—column-name—*,*—search-argument—*)—►►

函数参数

column name

列的名称。一列必须具有相关联的文本索引。可以使用管理命令 DB2TEXT CREATE INDEX 来创建文本索引。

search-argument

包含要搜索的术语的类型为 VARCHAR 的字符串。参见第 159 页的第 14 章，『搜索自变量的语法』。

注意


不能对为视图创建的文本索引使用 NUMBEROFMATCHES 查询。

SCORE

SCORE 标量函数对文本文档进行搜索，并为找到的每个文档返回一个分数值，该值指示搜索自变量是如何描述找到的文档的。

SCORE 将返回 DOUBLE 值。当搜索术语在文档中出现得更频繁时，文档的分数就会增大。

函数语法

►►SCORE(*—column-name—*,*—search-argument—*)

函数参数

column name

列的名称。一列必须具有相关联的文本索引。可以使用管理命令 DB2TEXT CREATE INDEX 来创建文本索引。

search-argument

包含要搜索的术语的类型为 VARCHAR 的字符串。参见第 159 页的第 14 章，『搜索自变量的语法』。

注意

不能对为视图创建的文本索引使用 SCORE 查询。

DB2EXT.TEXTSEARCH

除了存储过程搜索和 SQL 标量搜索函数之外，Net Search Extender 还提供了与存储过程看起来非常相似的两个 SQL 表值函数。

这两个表值函数都称为 `db2ext.textsearch`。它们之间唯一的区别就是：一个表值函数支持 `HIGHLIGHT` 函数，并且具有两个附加参数：`numberOfHits` 和 `hitInformation`。

注意，不能对具有复合主键的表使用表值函数。

有关使用 `HIGHLIGHT` 函数的信息，参见第 175 页的『`DB2EXT.HIGHLIGHT`』。

注意

仅当用户表存储在具有单个节点的表空间中时，才能在分布式 DB2 环境中使用表值函数。还必须确保使用 `DB2NODE` 环境变量连接至正确的节点。

函数语法

1. `db2ext.textsearch` without highlight support

```
db2ext.textSearch
(
  query           VARCHAR(4096),
  indexSchema     VARCHAR(128),
  indexName       VARCHAR(128),
  resultFirstRow  INTEGER,
  resultNumberRows INTEGER,
  primKeyBinding  <supported types>,// same type as primary key
)

return table
(
  primKey          <supported types>,// same type as primary key
  numberOfMatches  INTEGER,
  score            DOUBLE,
  totalNbResults   INTEGER
)
```

2. `db2ext.textsearch` with highlight support

```
db2ext.textSearch
(
  query           VARCHAR(4096),
```

```
indexSchema      VARCHAR(128),
indexName        VARCHAR(128),
resultFirstRow   INTEGER,
resultNumberRows INTEGER,
primKeyBinding   <supported types>,// same type as primary key
numberOfHits    INTEGER
)

return table
(
primKey          <supported types>,// same type as primary key
numberOfMatches  INTEGER,
score            DOUBLE,
totalNbResults   INTEGER
hitInformation   BLOB(20K)
)
```

函数参数

下面是一些输入参数。

query 有关附加信息，参见第 159 页的第 14 章，『搜索自变量的语法』。

indexSchema, indexName

标识要搜索的索引。有关更多信息，参见第 130 页的『CREATE INDEX』。

resultFirstRow

各个部分返回的查询结果列表。此参数描述查询结果列表的哪一行是要输入到表值函数的结果表中的第一行。该值应当大于等于零。

注意，数字 0 标识查询结果列表中的第一行。

resultNumberRows

此参数描述将查询结果列表中的多少行输入到了表值函数的结果表中，如果该值为 0，则意味着所有结果都需要返回。

注意：此参数与用来确定查询结果列表的最大大小的结果限制查询参数是不同的。

primaryKeyBinding

此参数的类型确定 **primaryKey Output** 参数的类型。如果为具有 **<type1>** 类型的主键的基本表创建了文本索引，则 **primaryKeyBinding** 也必须是 **<type1>** 类型。

另外，该参数还确定文本搜索的范围。如果 **primaryKeyBinding** 设置为 **NULL ("CAST(NULL as <type1>")**，则搜索范围将是存储在索引中的所有文档。或者，可以限制只对 **primaryKeyBinding** 绑定至的文档进行搜索。

例如，如果将 `primaryKeyBinding` 设置为 `CAST(5 as BIGINT)`，则限制只对 `BIGINT` 主键值为“5”的单个文档进行搜索。

注意，只支持下列类型的单个列主键：`SMALLINT`、`INTEGER`、`BIGINT`、`REAL`、`DOUBLE`、`VARCHAR FOR BIT DATA`、`DATE`、`TIME` 和 `TIMESTAMP`。

numberOfHits

此选项指定由 `db2ext.textsearch` 函数返回的匹配项的最大信息数。如果指定 0，则最多提供 1100 个匹配项的信息。这个过程可能很耗时间。

注意：只有构造 `db2ext.highlight` 函数必需的突出显示信息才需要此参数。

函数参数

下列返回值存储在临时表中，如果请求进一步的结果，则需要将该临时表连接至正常表。注意，仅当在 `select` 语句中请求了 `NUMBEROFMATCHES`、`SCORE`、`TOTALNUMBEROFRESULTS` 和 `HITINFORMATION` 时才会对这些函数求值。

primKey

找到的文档的主键。

numberOfMatches

`NUMBEROFMATCHES` 是一个 `INTEGER` 值，它指示每个文档产生了多少个匹配项。

score `Score` 将返回 `DOUBLE` 值。当搜索术语在文档中出现的频率提高时，文档的分数就会增大。

totalNumberOfResults

查询结果列表指示找到了多少个结果。注意，每一行都具有相同的值。

还要注意，当在查询中将 `STOP SEARCH AFTER` 或 `RESULT LIMIT` 与 `SCORE` 语法一起使用时，此数值不再可靠。

hitInformation

由 `db2ext.textsearch` 返回的匹配项信息对于进行突出显示处理是必需的。目前，此输出参数中大约可以包含 1100 个匹配项。如果匹配项数目超过了此阈值，则会忽略后续匹配项的匹配项信息。

注意，仅当指定了 `numberOfHits` 时才会返回此值。

使用法

借助 SQL 表值函数，可以按搜索存储过程搜索的相同方式来搜索视图。唯一的例外就是不需要共享内存，因此不需要激活索引。

SQL 表值函数

此函数主要供已经在存储过程搜索中使用了 SQL 查询的那些用户使用。但是，存在的限制就是只支持基本表的单列主键。

以下示例说明可以如何处理多列主键表:

```
select s.id from
db2ext.sample s, table (db2ext.textSearch(
    '"characteristics"',
    'DB2EXT',
    'COMMANDS',
    1,
    20,
    cast(NULL as INTEGER))) t
where s.id = t.primkey
```

在此示例中，必须首先对具有单个唯一键的此表创建视图，然后对此视图创建索引。

有关将 SQL 表值函数与 db2ext.highlight 函数配合使用的示例，参见第 176 页。

DB2EXT.HIGHLIGHT

使用 `db2ext.highlight` 函数来获取有关将文档限定为搜索结果的原因的信息。更准确地说，它可以用来：

- 获取匹配项
- 获取匹配项和周围的文本
- 在文档中的匹配项周围添加用户定义的突出显示标记。

注意，只能将 `db2ext.highlight` 函数与 `db2ext.textsearch` 表值函数配合使用。表值函数搜索索引，并将结果提供给 `HIGHLIGHT` 函数使用。

有关使用 `db2ext.textsearch` 函数的信息，参见第 171 页的『`DB2EXT.TEXTSEARCH`』。

函数语法

```
►—db2ext.highlight—————►
►(—document-content—,—hit-information—,—hit-processing-information—)————►
```

函数参数

下面是一些输入参数：

document content CLOB(100K)

只支持 `TEXT` 或 `XML` 格式的 `UTF8` 文档。要增加此值，参见第 119 页的『`DB2EXTHL`（实用程序）』。

hit information BLOB(20K)

包含匹配项信息的字符串。如果指定了 `numberOfHits` 参数的话，则此参数值是由 `db2ext.textsearch` 函数返回的。

hit processing information VARCHAR(1024)

此参数是选项值对的列表，它们之间用逗号“,”字符隔开，并将每个字符串字符用“ ”字符引起来。它指定应当如何对指定的文档处理突出显示。如果未指定任何选项，则将返回原始文档。

TAGS = ("STRING", "STRING")

此选项使用户能够指定要插入到文档中的匹配项前面和后面的标记。如果省略了此选项，则在文档中的匹配项前面和后面不会添加任何标记。

WINDOW_NUMBER = INTEGER

此选项指定突出显示函数应当返回文档的多少个部分（或窗

口)。每个窗口都包含一个或多个匹配项，每个窗口中的第一个匹配项确定返回至用户的文档的部分。这些匹配项可能具有也可能没有文本包围在匹配项周围。

如果省略此选项，则采用 0 作为缺省值，并且会返回包含开始和结束标记（如果指定的话）的整个文档。在此情况下，忽略了 WINDOW_SIZE 选项。

WINDOW_SIZE = INTEGER

此选项指定建议的窗口大小（以字节计）。此实际大小可能会根据匹配项数目、匹配项长度以及开始和结束标记大小的不同而有所变化。如果省略了该选项，则 0 为缺省值，并且将只返回周围没有文本的匹配项。

WINDOW_SEPARATOR = "STRING"

此选项指定用来将一个窗口与相邻窗口隔开的标记。如果省略了该选项，则 “...” 为缺省值。

FORMAT = "STRING"

此选项指定文档的格式。有效值为 XML 或 TEXT。如果省略了此选项，则采用 TEXT 作为缺省值。确保格式值与建立索引期间指定的值相同。

MODEL_NAME = "STRING"

此选项指定与所指定的 XML 文档相关的模型名。注意，如果 FORMAT 为 TEXT，则此选项将产生错误。

SECTIONS = ("section-name1", ..., "section-nameN")

对于 XML 文档，可以限制只突出显示相关部分。例如，可以在模型文件中定义它们。要指定这些部分，用逗号将一个或多个部分名隔开。如果省略了此选项，则对整个 XML 文档执行突出显示。注意，如果 FORMAT 为 TEXT，则忽略此选项。

函数参数

下面是一些返回参数。

CLOB(200K)

突出显示函数将返回包含由突出显示函数修改的文档部分的 CLOB 值。

使用法

以下示例说明了可以如何使用 HIGHLIGHT 函数：

```
select p.id,  
       p.title,  
       db2ext.highlight(p.content,
```



```

t.hitinformation,
'TAGS = ("<bf>", "</bf>"),
WINDOWS_NUMBER = 5,
WINDOWS_SIZE = 200,
WINDOW_SEPARATOR = "...",
FORMAT = "XML",
SECTIONS = ("section1-name", "section2-name"))

FROM patent p, table (db2ext.textsearch(
    "relational database systems",
    'DB2EXT',
    'TI_FOR_CONTENT',
    0,
    20,
    CAST(NULL as BIGINT),
    15)) t

WHERE p.id = t.primkey

```

使用超过 100 KB 的文档将导致 SQL 查询终止并且产生 SQL 错误（SQL 1476N 和 sql 错误 -433）。要避免发生此情况，则使用 db2exthl 命令来增加文档内容大小。有关信息，参见第 119 页的『DB2EXTHL（实用程序）』。

注意

特殊字符（例如“newline”）将按原样返回。

限制

- 只支持 XML 和平面文本文档
- 只有 UTF8 数据库受支持。对于二进制文件或 datalink 文档，需要确保文档是以 UTF8 编写的
- 不支持泰语文档
- 如果建立索引期间与查询时使用的文档格式不匹配，则 HIGHLIGHT 函数将返回无法预测的结果
- 将只突出显示在文档的文本部分找到的匹配项
- 突出显示函数只能与 db2ext.textsearch 函数配合使用
- 字符串值不能包含 " 字符

第 16 章 存储过程搜索函数

Net Search Extender 提供了一个存储过程搜索来返回预定义的结果表。结果表是在创建索引期间在高速缓存表部分中指定的。当需要按特定次序返回少量结果时使用存储过程搜索。

一个示例是这样一个因特网应用程序：它返回前 20 行，而其余行也以 20 行为增量来返回。

注意

仅当用户表存储在单节点表空间中时，才能在分布式 DB2 环境中使用存储过程函数。

还必须确保使用 DB2NODE 环境变量连接至正确的节点。

DB2EXT.TEXTSEARCH (对于存储过程搜索)

函数语法

```
db2ext.TextSearch(  
  
    IN      query          VARCHAR(4096),  
    IN      indexSchema    VARCHAR(128),  
    IN      indexName      VARCHAR(128),  
    IN      resultFirstRow INTEGER,  
    IN      resultNumberRows INTEGER,  
    IN      scoringFlag    INTEGER,  
    IN      searchTermCountsFlag INTEGER,  
    OUT     searchTermCounts VARCHAR(4096),  
    OUT     totalNumberOfResults INTEGER )
```

函数参数

下面是一些输入参数。

Query 有关进一步的信息，参见第 159 页的第 14 章，『搜索自变量的语法』。

indexSchema, indexName

标识要搜索的索引。参考第 130 页的『CREATE INDEX』。

resultFirstrow

采用多个部分来返回查询结果列表。该参数描述查询结果列表的哪一行是要放置到存储过程的结果集中的第一行。查询结果列表中的第一行是用数字 0 标识的。

resultNumberRows

此参数描述将查询结果列表中的多少行放置到存储过程的结果集中。

这不会与查询中的 "result limit" 表达式混淆，该表达式确定查询结果列表的最大标志大小。

该值应当大于等于零。其中零表示需要返回所有结果。

注意

如果请求了一个较大的结果集，则应确保提供了临时用户表空间。如果没有提供临时表空间，则创建一个表空间。以下示例将在 UNIX 平台上创建一个表空间：

```
db2 "create user temporary tablespace tempts managed by system  
    using ('/work/tempts.ts')"
```

scoringFlag

0 表示没有计算分数，而 1 表示计算了分数。如果请求了计算分数，则添加返回了分数值的行。

searchTermCountsFlag

此参数将控制 `searchTermCounts` 处理。如果 `searchTermCountsFlag` 为 0，则不计算 `searchTermCounts`。

函数参数

下面是一些输出参数。

searchTermCounts

索引中每个搜索术语查询出现的次数。这些计数按照查询中搜索术语的次序作为用空格隔开的列表来返回。

有关信息，参见 **searchTermCountsFlag**。

totalNumberOfResults

在查询结果列表中的总结果数。

还要注意，当在查询中将 `STOP SEARCH AFTER` 或 `RESULT LIMIT` 与 `SCORE` 语法一起使用时，此数值不再可靠。

使用法

结果集中由存储过程返回的列是由 `DB2TEXT CREATE INDEX` 命令的 `RESULT COLUMNS` 选项给定的。如果 `scoringFlag=1`，则会添加类型为 `double` 的一列。此列包含 `SCORE` 值。

使用下列选项来提高与第一个查询具有相同字符串的第二个查询的性能。注意，这必须是在不需要 `totalNumberOfResults` 的另一个游标窗口中：

- 如果不需要 `SCORE`，则添加以下语法：`STOP SEARCH AFTER x DOCUMENTS`，其中 `x` 是 `resultFirstRow + resultNumberRows`。
- 如果需要 `SCORE`，则添加以下语法：`STOP SEARCH AFTER y DOCUMENTS`，其中 `y` 等于第一个查询中的 `totalNumberOfResults`。

为了确保连接至正确的节点以进行搜索，可能必须设置 `DB2NODE` 环境变量。

对于 UNIX，使用以下命令：

```
export DB2NODE=<no>
```

注意，所有物理节点都具有同步时间很重要。

对于 Windows，使用：

存储过程搜索函数

```
set DB2NODE= <no>
```

注意

与实例所有者标识不同的受防护用户标识不使用分区数据库。

具有 SQL 查询的存储过程

此存储过程允许将文本查询的结果与附加 SQL 查询进行组合。其使用范围仅限于对具有单个键列的表 / 视图创建的索引。

```
Textsearch(
    IN          query          VARCHAR(32000),

TextSearchSql(
    IN          query          VARCHAR(32000),
    IN          indexSchema    VARCHAR(128),
    IN          indexName      VARCHAR(18),
    IN          resultFirstRow BIGINT,
    IN          resultNumberRows BIGINT,
    IN          scoringFlag    INTEGER,
    IN          sqlQuery        VARCHAR(32000),
    INOUT       searchTermCounts VARCHAR(32000),
    OUT         totalNumberOfResults BIGINT)
```

输入参数

有关信息，参见 TextSearch 存储过程以及下列内容：

sqlQuery

指定存储过程返回的结果表中列和行的 SQL 语句。对于未排序的查询（排序标志 = 0），该语句具有以下结构：

```
select ...from ... where ...
```

where 子句（属于引用表或对其创建索引的视图的 from 子句）必须在 SQL 允许 IN 谓词的位置包含 “<keycolumn>in (%s)”。

对于已排序的查询（排序标志 = 1），该语句具有以下结构：

```
select ...from ... where ...
```

select 子句必须在 SQL 允许表达式的位置包含列引用 “RSCORE”。from 子句必须在 SQL 允许表引用的位置包含 “%s”。where 子句（属于引用表或对其创建索引的视图的 from 子句）必须在 SQL 允许 IN 谓词的位置包含 “%s”。

输入 / 输出参数

有关信息，参见 TextSearch 存储过程。

输出参数

有关信息，参见 TextSearch 存储过程。

存储过程搜索函数

输出结果集

有关信息，参见 `TextSearch` 存储过程。

错误处理

有关信息，参见 `TextSearch` 存储过程。

第 17 章 结构化文档支持

结构化文档由文档模型和文档文件定义组成。

文档模型

文档模型主要控制需要对文档结构的哪些部分建立索引以及如何对它们建立索引。其目的为:

- 标识应在源文档中区别出来的文本字段
- 确定这样的文本字段的类型
- 向文本字段指定字段名

当文档模型将文本标识为属于文本字段时, 将认为该文本是文档的文本内容的一部分, 并将抽取一些术语并将它们存储在索引中。

文档模型的元素随用于该文档格式的解析器的不同而变化:

- 对于 HTML 格式, 文档模型使用 HTML 标记名来定义应对哪些标记建立索引以及如何处理元标记信息。
- 对于 XML 格式, 没有预定义的标记集, 所以文档模型必须首先定义哪些标记是有用的。对于名称相同的 XML 元素, 还可以根据它们嵌入哪些其它元素中来区分它们。
- 对于 GPP (通用解析器) 格式, 文档模型与解析器进行更进一步的交互, 原因是它必须确定文本字段的边界。此处字段定义必须指定用于检测字段边界的字符串。
- 对于 Outside-In 格式, 文档模型使用类似 HTML 标记名的标记来定义应对哪些标记建立索引以及如何处理元标记信息。注意, Outside-In 过滤格式也称为 INSO。

有关信息, 参见相关“定义文档模型”一节。

有关“文档类型定义”(DTD)形式的文档模型语法的信息和文本字段限制, 参见第 253 页的附录 G, 『文档模型参考』。

缺省文档模型

对于 HTML 和 XML 文档, Net Search Extender 提供在没有定义文档模型的情况下使用的缺省文档模型。对于结构化纯文本文档, 必须提供并指定文档模型。

如果使用缺省文档模型之一，则：

- 将对所有字段建立索引，并且不会抽取任何特殊信息（如元信息）。
 - 对于 HTML，将相应标记的名称分配给每个字段。
 - 对于 XML，生成的字段名是完整的标记路径，例如 `/play/role/name`。
- 不会对数字属性建立索引（原因是未在缺省文档模型中定义任何数字属性）。

表 6. 受支持文档格式的缺省文档模型的行为

文档类型	缺省文档模型的行为
HTML	接受以下内容作为文本字段：<a> <address> <au> <author> <h1><h2> <h3> <h4> <h5> <h6> <title>。 字段名为标记名，例如 “address”。
XML	接受所有标记作为文本字段。 字段名为 Xpath 注释中的标记路径名，例如 “/play/title”。
结构化纯文本（GPP）	没有缺省文档模型。
Outside-In（INSO）	接受由 Outside-In 过滤器返回的第 196 页的『元素参数』中显示的文档属性作为文本字段。字段名是 Outside-In 使用的文档属性的名称，例如：“SCCCA_TITLE”。没有任何属性支持。

对于每种类型的文档，都定义了文档模型。因为模型各不相同，所以为每个模型提供了示例和解释。

注意

尽管缺省文档模型的确正确地处理文档，但为了更好地建立索引和搜索，您应该定义自己的文档模型。

借助缺省文档模型，对文档的文本全部建立索引，而不管它是否为文本字段的一部分。这意味着非限制文档搜索包括对该文本的搜索。

为结构化纯文本文档定义文档模型

以下是通用（GPP）结构化纯文本文档的示例：

```
[head]Handling structured documents
[/head]
[year]2002
```

```
[/year]
[abstract]This document describes the concept of structured documents
and the use of document models to...
[/abstract]
```

以下是 GPP 文档模型的示例:

```
<?xml version="1.0"?>
<GPPModel>

  <GPPFieldDefinition
    name="Head"
    start="[head]"
    end="[/head]"
    exclude="YES" />

  <GPPFieldDefinition
    name="Abstract"
    start="[abstract]"
    end="[/abstract]"
    exclude="NO" />

  <GPPAttributeDefinition
    name="year"
    start="[year]"
    end="[/year]"
    type="NUMBER" />

</GPPModel>
```

- This is the start of text field

- This is the end of a text field

- This is the start of a document attribute

- This is the end of a document attribute

第一行 `<?xml version="1.0"?>` 指定使用 XML 标记编写文档模型。注意, 此模型不是为 XML 格式文档编写的。

每个字段都是在 `GPPFieldDefinition` 或 `GPPAttributeDefinition` 标记内定义的, 该标记包含元素参数。

所有定义都必须包含在 `<GPPModel>` 标记内。

元素参数

以下是文档模型元素的参数:

- name** 对每个定义的文本字段或文档属性指定名称。这些名称使您能够将搜索查询限制为特定文本字段或文档属性的内容。使用上面的示例, 您可在名为 `Abstract` 的文本字段中搜索包含单词 `structure` 的文档。
- start** 代码页 UTF-8 中的边界字符串, 它标记文本字段或文档属性的开始。指定边界字符串没有任何规则; 它们可以是任意 UTF-8 字符串。以下是一些示例: `start="introduction:"`、`start="note!"` 和 `start="$$..."`。

必须使用标准 XML 转义字符（“<”表示“<”，而“&”表示“&”）来指定不可打印字符和特殊 XML 字符“<”和“&”。

end 可选。代码页 UTF-8 中的边界字符串，它标记文本字段或文档属性的结束。如果不指定结束标记，则假定找到的下一个开始标记作为该字段的结束。如果找不到任何后继开始标记，则该字段将一直扩展至文档结尾，并且不会再标识任何其它字段。

type 文档属性的类型必须始终是“NUMBER”。该参数不适用于字段定义。

exclude

YES 或 NO。一个参数，它确定是否应排除字段中的文本并且不对其建立索引。此参数不适用于属性定义。在该示例中，将排除字段定义“head”，但是将包括定义“abstract”。

限制:

- 一定不能有两个具有相同开始标记的字段定义或属性定义。但是，字段定义和属性定义可以有相同的开始标记和结束标记。
- 一个开始标记一定不能恰好是另一个开始标记的前缀。例如，不能有一个开始标记是“author”而另一个开始标记是“authority”的情况。
- 开始标记和结束标记一定不能是空的字符串。

有关“文档类型定义”的信息，参见第 253 页的『文档模型的 DTD』。

有关附加限制，参见第 256 页的『文本字段和文档属性的限制』。

对 GPP 文档建立索引时的情况

通用解析器将扫描文档以查找开始边界字符串之一。当它找到一个开始边界字符串时，它将分析后继字段直到找到相应的结束边界字符串为止。

然后将根据定义术语对字段的内容建立索引，即，作为文本字段或文档属性。如果文本字段和文档属性具有相同的开始和结束边界字符串，则将把字段的内容同时作为文本字段和文档属性来对其建立索引。

不允许字段的嵌套；如果在到达结束边界字符串前发现新的开始边界字符串，则会将新的开始边界字符串解释为普通文本。

如果未找到相应的结束边界字符串，则假定该字段扩展至文档结束，并将报告适当的原因代码。

如果在文档模型中未指定结束边界字符串，则新的开始边界字符串指示先前字段的结束。

为 HTML 文档定义文档模型

HTML 解析器将文本转换为代码页 UTF-8。它执行 HTML 标记识别，并将它们归类为下列标记类：

- 要忽略的已标记信息，如字体信息
- 提供位置信息的标记，如表示新段落的 `<p>`；
- 提供结构信息的标记，如 `<Title>`

它将识别 HTML 4 中定义的所有字符实体引用（如 “ä”（ä））并将它们解析为 UTF-8 中相应的代码点。

它将识别元标记并分析元标记文本。

以下是 HTML 文档的示例：

```
<HTML>
<HEAD>
<META NAME="year" CONTENT="2002">
<TITLE> The Firm </TITLE>
</HEAD>
<BODY>
<H1>Synopsis</H1>;
```

```
<H1>Prologue</H1>;:
</BODY>
```

以下是 HTML 文档模型的示例：

```
<?xml version="1.0"?>
<HTMLModel>

  <HTMLFieldDefinition
    name="subtitle"
    tag="title"
    exclude="YES" />

  <HTMLFieldDefinition
    name="header1"
    tag="h1"
    exclude="YES" />

  <HTMLAttributeDefinition
    name="year"
    tag="meta"
    meta-qualifier="year"
    type="NUMBER" />

</HTMLModel>
```

- This is the start of text field

- This is the end of the text field

- This is the start of the document attribute

- This is the end of the document attribute

第一行 `<?xml version="1.0"?>` 指定使用 XML 标记编写文档模型。注意，此模型不是为 XML 格式文档编写的。

每个字段都是在 `HTMLFieldDefinition` 或 `HTMLAttributeDefinition` 标记内定义的，该标记包含元素参数。

所有文本字段定义都必须包含在 `<HTMLModel>` 标记内。

元素参数

以下是文档模型元素的参数：

name 对每个定义的文本字段或文档属性指定名称。这些名称使您能够将搜索查询限制为特定文本字段或文档属性的内容。使用上面的示例，您可在名为 `subtitle` 的文本字段中搜索包含单词 `firm` 的文档。

tag 标识其开始和（暗示的）结束标记标识文本字段或文档属性的元素。该名称的元素内的文本组成已定义字段的内容。

忽略标记的大小写。

使用上面的示例，将任何 `H1` 标记之后的文本作为字段“`header1`”的一部分建立索引。在这种情况下，将对“`synopsis`”和“`prologue`”建立索引。

meta-qualifier

此标记必须与 **tag** 元素配合使用。通过指定 `tag="meta"`，将抽取与 `meta-qualifier` 相匹配的内容的值。

在 HTML 文档示例中，元标记具有下列元素：

```
<META NAME="Author" CONTENT="J. Grisham">
```

在文档模型示例中，则为 `meta-qualifier="author"`。因此，“`J.Grisham`”内容将作为字符串属性“`author`”的值建立索引。

type 文档属性的类型必须是“`NUMBER`”。该参数不适用于字段定义。

exclude YES 或 NO。一个参数，它确定是否应排除字段定义中的文本并且不对其建立索引。此参数不适用于属性定义。在该示例中，将排除字段定义“`header1`”，但是将包括定义“`subtitle`”。

将对文档的所有其它文本建立索引，但不作为任何字段的一部分。

有关“文档类型定义”的信息，参见第 253 页的『文档模型的 DTD』。

有关限制，参见第 256 页的『文本字段和文档属性的限制』。

为 XML 文档定义文档模型

Net Search Extender 不会尝试检测 XML 文档的代码页。CREATE INDEX 期间指定的 CCSID，或者如果未指定它的话，则为 DB2 代码页。

以下是 XML 文档的示例:

```
<?xml version="1.0"?>
<purchaseOrder orderDate="2001-01-20">
  <shipAddress countryCode="US">                                [1]
    <name>Alice Smith</name>                                    [2]
    <street>123 Maple Street</street>
    <city>Mill Hill</city>
    <state>CA</state>
    <zip>90999</zip>
  </shipAddress>
  <item partNo="123" quantity="1">
    <name>S&B Lawnmower Type ABC-x</name>                        [3]
    <price>239.90</price>
    <shipDate>2001-01-25</shipDate>
  </item>
  <item partNo="987" quantity="1">                                [3]
    <name>Multifunction Rake ZYX</name>
    <price>69.90</price>
    <shipDate>2001-01-24</shipDate>
  </item>
</purchaseOrder>
```

以下是 XML 文档模型的示例:

```
<?xml version="1.0"?>
<XMLModel>

  <XMLFieldDefinition                                          [1]
    name="addresses"
    locator="/purchaseOrder/shipAddress"
    exclude="no" />

  <XMLFieldDefinition                                          [2]
    name="customerName"
    locator="//shipAddress/name"
    exclude="yes"/>

  <XMLAttributeDefinition                                     [3]
    name="partNumber"
    type="NUMBER"
    locator="/purchaseOrder//item/@partNo" />

</XMLModel>
```

第一行 `<?xml version="1.0"?>` 指定使用 XML 编写模型。每个字段都是在 `XMLFieldDefinition` 或 `XMLAttributeDefinition` 标记内定义的，它包含元素参数。

注意，所有文本字段定义都必须包含在 `<XMLModel>` 标记内。有关限制，参见第 256 页的『文本字段和文档属性的限制』。

元素参数

以下是文档模型元素的参数:

name 对每个定义的文本字段或文档属性指定名称。这些名称使您能够将搜索查询限制为特定文本字段或文档属性的内容。

可在名称中使用下列变量之一。该变量将由源文档中的匹配元素生成的字符串来替换。

Variable	值
\$(NAME)	与 XPath 路径相匹配的 XML 元素的实际限定名 (QName)。
\$(LOCALNAME)	与 XPath 相匹配的 XML 元素的实际本地名 (不带前缀)。
\$(PATH)	实际绝对路径，以与 XPath 相匹配的 XML 元素的斜杠和标记的序列组成。

type 文档属性的类型必须是 “NUMBER”。该参数不适用于字段定义。

locator

XPath 语言中的表达式，它们选择要用作搜索字段的源文档的各个部分。这些定位器来自示例。有关进一步的信息，参见第 254 页的『定位器 (XPath) 表达式的语义』中的语法。

purchaseOrder salesOrder	所有 purchaseOrder 元素和 salesOrder 元素
shipAddress	所有 shipAddress 元素
*	所有元素 (它是 child::* 的缩写 – 有关进一步的信息，参见语法)
name/item	具有 name 父代的所有 item 元素
purchaseOrder//item	具有 purchaseOrder 祖先的所有 item 元素

/ 根节点

comment() 所有注释节点

processing-instruction()
所有处理指令

attribute::* (or @*)
所有属性节点

NCName

不包含冒号的 XML 名称

QName

前面加上 NCName: (NCName 后跟冒号) 的 NCName, 如下所示: NCName:NCName

文字为括在单引号或双引号内的字符串。有关终端标记的精确定义, 参见 XML 建议。

XPath 定位器与“XML 样式表语言变换”(XSLT)模式相似。它们正是由不包含任何谓词或函数“id”和“key”或节点测试“text()”和“node()”的 XSLT 模式的子集组成。

ignore

YES 或 NO。使用该参数来建立定位器的异常。

有时您可能想要指定常规定位器(例如*)以与想要对其建立索引的节点相匹配。但是您还可指定不应对与较特定定位器相匹配的某些节点建立索引。

为此, 为在建立索引期间要忽略的节点的较特定定位器加上字段定义。于是就对此定位器指定了比常规定位器更高的优先级, 并指定 ignore="yes"。这将指示索引程序一定不要为匹配节点生成字段信息。

注意, 当这种被忽略节点嵌入在字段生成节点中时, 将对被忽略节点的内容建立索引, 原因是它也属于字段生成节点的内容。

priority

-1 和 +1 之间的实数, 它指定要给予由特定定位器找到的定义的优先级。

如果不指定优先级, 将使用缺省优先级:

- 将由 | 分隔的多个备用项视作一组定义, 每个备用项一个定义。
- 与单个名称相匹配的定位器; 即, 下列任一格式的定位器具有缺省优先级 0:
 - ChildOrAttributeAxisSpecifier QName
 - ChildOrAttributeAxisSpecifier processing-instruction(Literal))

- 格式为 ChildOrAttributeAxisSpecifier NCName:* 的定位器具有缺省优先级 -0.25。
- 格式为 ChildOrAttributeAxisSpecifier NodeTest 的其它定位器具有缺省优先级 -0.5。
- 任何其它定位器具有缺省优先级 0.5。

注意定位器越具体，缺省优先级越高。例如，非特定定位器 * 给予找到的定义低优先级，而名称较特定的定位器给予较高的优先级。

另外，注意当一个节点与多个定位器相匹配时，您可以通过对它们指定优先级来确定选择哪个定义。将选择具有最高优先级的定义。如果两个定义具有相同的优先级，将选择最新的定义。

此冲突解决与在“XML 样式表语言变换”（XSLT）中使用的完全相同。

exclude

YES 或 NO。一个参数，它确定是否应排除字段定义中的文本并且不对其建立索引。此参数不适用于属性定义。

在该示例中，将排除字段定义“customerName”，但是将包括定义“addresses”。

对 XML 文档建立索引时的情况

下表显示将哪些内容放入索引。

表 7. 文本索引中的条目

字段名	已索引文本	
addresses	123 Maple Street Mill Hill CA 90999	[1]
customerName	Alice Smith	[2]
partNumber	123 987	[3]

注意在 [1] 中 shipAddress 元素的文本在字段名 addresses 下建立索引，未对 name 元素（Alice Smith）建立索引。这是因为 name 元素本身在文档模型中，并且在字段名 customerName [2] 下建立索引。这意味着尽管在文档模型中允许嵌入元素，但是并不将它们作为嵌入文本字段的一部分建立索引；而是单独对其建立索引。

字段的内容由下列规则确定：

- 对于其定位器与注释或处理指令相匹配的字段，字段内容为实际注释文本或处理指令文本。
- 对于与 XML 元素或根节点相匹配的字段，字段内容由来自任何嵌入元素（与其它字段相匹配的元素除外）的任何文本组成。

文档必须包含格式良好的 XML，但对于要在 XML 文档中指定的 DTD，则不必如此。不进行 DTD 验证或实体解析；Net Search Extender 只针对文档模型来匹配 XML 文档。

有关“文档类型定义”的信息，参见第 253 页的『文档模型的 DTD』。

有关限制，参见第 256 页的『文本字段和文档属性的限制』。

为 Outside-In 过滤的文档定义文档模型

Outside-In 格式的文档模型允许您将由给定标记集标识的结构化元素映射至 NSE 文本字段和文档属性，在这一点上它们与 HTML 文档模型很相似。假定有一组 Microsoft Word 文档并且您想要将文档属性“title”、“subject”和“keyword”作为字段建立索引，并且将文档特性“author”和“category”作为文档属性建立索引。Outside-In 文档模型的以下示例将完成此映射：

```
<?xml version="1.0"?>
<INSOModel>

<INSOFieldDefinition
name="title"
tag="SCCCA_TITLE"/>

<INSOFieldDefinition
name="title"
tag="SCCCA_SUBJECT"/>

<INSOFieldDefinition
name="title"
tag="SCCCA_KEYWORDS"/>

<INSOAttributeDefinition
name="author"
tag="SCCCA_AUTHOR"
type="STRING"/>

<INSOAttributeDefinition
name="category"
tag="SCCCA_CATEGORY"
type="STRING"/>

</INSOModel>
```

元素参数

以下是文档模型元素的参数:

- name** 对文本字段或文档属性指定的名称。对每个字段定义指定字段名, 并对每个属性定义指定属性名。查询可通过使用这些名称将搜索限制为某个文本字段的内容并可搜索具有某个属性的文档。
- tag** 标识一个标记, 其开始和结束或暗示结束元素标记文本字段或文档属性。该名称的元素内的文本组成已定义字段或属性的内容。忽略标记的大小写。下面描述了可能的值。
- type** 文档属性的类型可以是“NUMBER”、“DATE”或“STRING”。此参数不适用于字段定义。
- exclude** “YES”或“NO”。一个参数, 它确定是否应排除字段定义内的文本。如果字段定义具有参数 `exclude="YES"`, 则不对那些文本字段建立索引。此参数不适用于属性定义。

Outside-In 文档模型由字段和 / 或属性定义组成, 它们每一个都定义一个名称和一个标记。对于属性定义, 还需要一个类型, 而字段定义具有可选“exclude”标志。对于 HTML 模型, 这种定义的名称属性定义将向其映射文档部分的 NSE 字段或属性的名称。它可以是任意 UTF-8 文本字符串。有关附加信息, 参见 Outside-In Content Access Specification V7.5。

有关与 Outside-In 开始、结束和文档属性标记相关的标记属性的可能的值列表, 参见第 257 页的『Outside-In 标记属性值』以获取进一步的信息。

对 Outside-In 文档建立索引时的情况

缺省情况下, 将所有文本作为不属于任何字段的内容建立索引。每当出现在文本流中的开始标记与当前活动的文档模型中的定义术语匹配时, 就将根据该定义术语来处理在开始标记与其相应结束标记之间的文本。例如, 作为已索引字段和已排除字段和 / 或属性来处理。

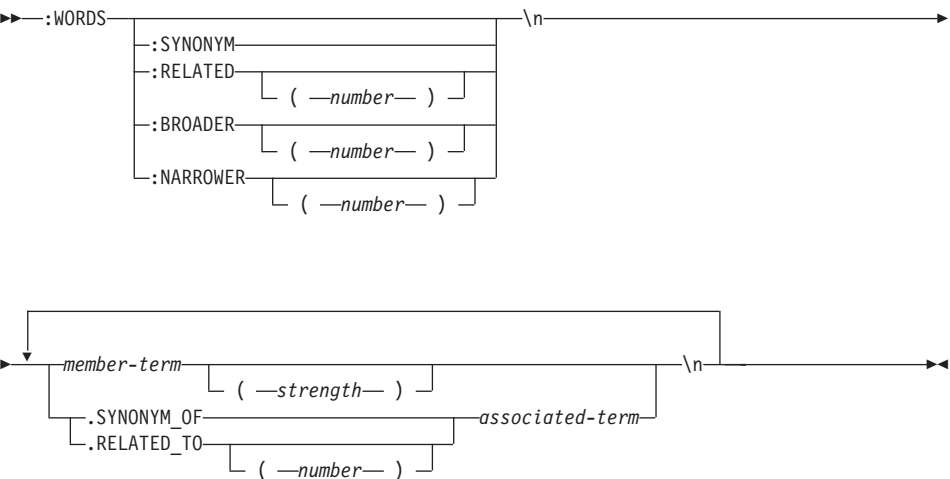
如果不存在匹配定义, 则将忽略开始标记及其相应的结束标记。

Outside-In 过滤器自动识别文档的格式和代码页, CCSID 规范不起任何作用。如果 **Outside-In** 过滤器无法确定正确的格式和代码页, 则会将文档视作 unicode (UTF-16) 文件。这与特定于 Outside-In 的文档类型 FI_UNICODE 相对应。

第 18 章 同义字支持

以下是每个定义组的语法:

同义字定义的语法



注意, \n 不是语法的一部分, 而是表示同义字定义文件中一行的结束。

可以在同义字定义文件中插入注释行, 如下所示:

```
# my comment text
```

:WORDS

开始一组相关词语的关键字。

:SYNONYM,

:RELATED [(number)],

:BROADER [(number)],

:NARROWER [(number)]

一个关系名。

关系名由关系类型和数字组成。如果省略了数字, 则假定它为零, 这是系统提供的关系名。:SYNONYM 始终是系统提供的关系名。

以冒号开头的关系名 (例如, :SYNONYM) 在彼此相关的词语列表前面添加同一关系。例如:

```
:WORDS
:SYNONYM
  air steward
  cabin staff member
  flight attendant
```

member-term

要包括在同义字典中的一个术语。

- 最大长度为 64 个字节（对于代码页 UTF-8，最大长度为 42 个字节）。
- 同一字母的单字节字符和双字节字符被认为是相同的。
- 不区分大小写字符。
- 术语可以包含空格字符。
- 不能使用单字节字符句点 “.” 或冒号 “:”。

如果不要同义字查找包括与查找的术语关系不大的词语，则此参数就非常有用。强度是 1 到 100 之间的数值。缺省值为 100。

.SYNONYM_OF, .RELATED_TO [(number)]

一个关系名。关系名由关系类型和数字组成。如果省略了数字，则假定它为零，这是系统提供的关系名。关系名 .SYNONYM 始终是系统提供的关系名。

以句点开头的关系名（例如，.SYNONYM_OF）定义一个词语与另一个词语之间的关系。例如：

```
:WORDS
  air steward
  .SYNONYM_OF cabin staff member
  .SYNONYM_OF flight attendant
```

可选的 *number* 标识用户定义的关系。这必须是整个同义字定义文件中的唯一数字（当前为 1 到 128）。例如：RELATED_TO(42)。

如果想要在应用程序中使用同义字关系的符号名称而不是关系名和数字，则应用程序必须处理名称至数字的映射。例如，如果将关系 *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)。

associated-term

必须在每个相关联的术语前面添加关系名。相关联的术语根据指定的关系来与每个成员术语相关。如果所有成员术语都互相相关，则可以使用成员关系来指定此参数。

- 最大长度为 64 个字节（对于代码页 UTF-8，最大长度为 42 个字节）。
- 同一字母的单字节字符和双字节字符被认为是相同的。
- 不区分大小写字符。

- 术语可以包含空格字符。
- 不能使用单字节字符句点 “.” 或冒号 “:”。

以下是相关联术语的一个示例:

```
:WORDS:SYNONYM  
  reject  
  decline  
    RELATED_T0(1) accept
```

第 3 部分 附录

附录 A. 迁移

已将 DB2 Net Search Extender V 8.1.x 作了大范围的变化以包括 DB2 Text Information Extender V7.2 的搜索接口和函数。相应地，存在三个迁移选项：

- 从 Text Information Extender V7.2 移至 Net Search Extender V8.1.x
- 从 Net Search Extender V7.2 移至 Net Search Extender V8.1.x
- 从 Net Search Extender V8.1 移至 Net Search Extender V8.1.x

注意

有关最新的迁移信息，检查 CD-ROM 和 DB2 Net Search Extender Web 站点上的 `release.txt` 文件。

从 Net Search Extender V8.1 移至 Net Search Extender V8.1.x

当从 Net Search Extender V8.1 移至 Net Search Extender V8.1.x 时不存在任何必要的预先步骤。因此，您可以除去旧的安装并安装 Net Search Extender 的最新版本。

对于 UNIX 和 Windows，使用 `db2iupdt` 以将 DB2 实例从 V8.1 迁移至 V8.1.x。然后使用新的 `db2extmdb` 工具迁移此实例的所有启用数据库。调用者必须是实例所有者，语法为：

```
db2extmdb <database name>
```

当迁移在运行时，不对具有文本索引的用户表作出任何更改。

注意事项和建议

迁移步骤记录在以下文件中：

```
<os-dependent>/sql1lib/db2ext/db2extm <database-name>.log
```

在调用 `db2extmdb` 程序之前，用户应对所有索引目录和数据库进行备份。

从 Net Search Extender V7.2 移至 Net Search Extender V8.1.x

如果已在使用 Net Search Extender V7.2，并且不需要所有新的功能部件，则只需要通过安装 CD-ROM 上（或通过下载）提供的 Net Search Extender V7.2 兼容性接口继续使用旧接口。有关进一步信息，参见 `release.txt`。要使用新的 Net Search Extender 功能，需要更改管理脚本以便与已改变管理语法相对应并将搜索语法更改为新搜索接口。

注意：从 Net Search Extender V7.2 迁移至 Net Search Extender V8.1.x 不存在任何自动方法。

还要注意：建议不要使用这些兼容性接口，它们在将来的发行版中不可用。

从 Text Information Extender V7.2 移至 Net Search Extender V8.1.x

当从 Text Information Extender V7.2 移至 Net Search Extender V8.1.x 时不存在任何必要的预先步骤。因此，您可以除去旧的安装并安装 Net Search Extender 的最新版本。

对于 UNIX 和 Windows，使用 `db2iupdt` 以将 DB2 实例从 V7.2 迁移至 V8.1.x。然后使用新的 `db2extmdb` 工具迁移此实例的所有启用数据库。调用者必须是实例所有者，语法为：

```
db2extmdb <database name>
```

在第一步中，程序将与迁移有关的所有 `db2ext` 管理信息收集到名为 `DB2EXT.TMIGRATION` 的新表中。在该表中，每个文本索引都表示为单独的一行。迁移信息表将保留至数据库成功迁移为止，不应由用户删除它。

如果发生错误，则修正错误并再次调用 `db2extmdb`。

文本索引迁移期间，发生不同的处理。这取决于“日志表”的状态。如果日志表是空的，则将迁移索引，这应该是最快的方法。如果日志表不是空的，则不能确保与数据库完全一致的状态，并且需要重新创建索引。此过程可能要花相当长的时间。

当迁移在运行时，不对具有文本索引的用户表作出任何更改。

建议

在调用 db2extmdb 程序之前，用户应该对所有索引目录和数据库进行备份，并且验证用于创建版本 7.2 文本索引的所有文档模型文件是否仍然存在，并且它们是可用于读取的。

注意事项和建议

迁移步骤记录在以下文件中：

```
<os-dependent>/sql1lib/db2ext/db2extm <database-name>.log
```

对于 HP-UX 或 Linux，此步骤不是必需的，原因是 Text Information Extender 在这些平台上不可用。

还要注意，DB2 Text Information Extender V7.2 不再可用。

附录 B. 使用大量内存

对于下列平台，将高速缓存用于存储过程搜索需要大量内存和不同的内存需求：

- AIX
- Windows
- Solaris 操作环境
- Linux
- HP-UX

AIX (32 位和 64 位)

配置系统限制：

- 使用命令 `ulimit -a` 检查系统限制
- 如果具有不属于“不受限制”的值，则使用下列步骤：
 - 作为 `root` 用户登录。
 - 备份 `/etc/security/limits` 文件，然后编辑该文件以提高硬限制。
 - 对于所使用的 DB2 实例所有者，将所有值设置为“不受限制”（值为 `-1`）。

配置共享内存限制：

- 在 AIX 上，不需要配置共享内存限制。

配置交换空间：

- 使用 `lsattr -E -l sys0` 命令获取系统 RAM 大小
- 使用 `lspv -a` 命令获取交换空间大小。
- 将交换空间大小设置为至少是系统的 RAM 数量的 1.5 至两倍，或者使用在 `CREATE INDEX` 命令中提供的 `MAXIMUM CACHE SIZE` 参数。使用 `SMIT` 实用程序来选择较大的数目。

注意，在 AIX 上，最大高速缓存大小限制为 1536 MB（1.5 GB = 1610612736 个字节）。

Windows (32 位)

调整页面调度文件的大小：

使用大量内存

- 将 Windows 虚拟内存页面调度文件大小设置为至少是系统的 RAM 数量的 1.5 至两倍，或者使用在 CREATE INDEX 命令中提供的 MAXIMUM CACHE SIZE 参数。选择较大的那一个数目。有关更改页面调度文件大小的信息，参见 Windows 文档。

注意，在 Windows 上，最大高速缓存大小限制为 1000 MB (1 GB = 1073741824 个字节)。

Solaris 操作环境 (32 位和 64 位)

配置系统限制:

- 使用 `ulimit -a` 命令来检查系统限制
- 然后，执行下列步骤:
 - 作为 root 用户登录。
 - 备份 `/etc/system` 文件，然后编辑该文件以提高硬限制。
 - 添加或检查是否将下列行设置为至少是所显示的最小值:
`rlim_fd_cur -> 缺省值 64, 建议值 >= 1024`
`rlim_fd_cur_max -> 缺省值 1024, 建议值 >= 4096`

配置共享内存限制:

- 使用 `sysdef -i` 命令来检查当前设置
- 编辑 `/etc/system` 文件，以使用 `set shmsys:shminfo_shmmax=0xffffffff` 来设置共享内存大小限制

可能还必须增大下列参数值:

```
set shmsys:shminfo_shmmni=512
set shmsys:shminfo_shmseg=128, 然后重新引导系统。
```

配置交换空间:

- 使用 `/usr/sbin/prtconf` 命令获取系统 RAM 大小
- 使用 `swap -l` 命令获取交换空间大小。
- 将交换空间大小设置为至少是系统的 RAM 数量的 1.5 至 2 倍，或者使用在 CREATE INDEX 命令中提供的 MAXIMUM CACHE SIZE 参数。选择较大的那一个数目。

有关如何添加交换空间的信息，参考 Solaris 系统文档。

注意，在 Solaris 上，最大高速缓存大小限制为 2000 MB (2 GB = 2147483647 个字节)。

Linux (32 位)

检查有关特定内核参数的 DB2 文档。

要查看当前共享资源限制，使用 `ipcs -l`。要检查系统限制，使用 `ulimit -a` 命令。

HP-UX (32 位和 64 位)

将 `shmmx` 参数设置为 134217728 或物理内存（以字节计）的 90%（取较高值）。

例如，如果在系统中具有 196 MB 物理内存，则将 `shmmx` 设置为 184968806 ($196 * 1024 * 1024 * 0.9$)。

附录 C. Net Search Extender 信息目录

DB2 Net Search Extender 将有关缺省值、配置、文本索引和格式的重要信息存储在目录表中。要查看此信息，可对表查询一些视图。

下列视图和表反映系统的当前配置：

- 数据库级别信息视图：
 - db2ext.dbdefaults
 - db2ext.proxyinformation 表
- 索引级别信息视图：
 - db2ext.textindexes
 - db2ext.textindexformats
 - db2ext.indexconfiguration

注意，由于兼容性原因，DB2 Text Information Extender 视图仍然可用。这些视图包括 db2ext.textcolumns、db2ext.models 和 db2ext.formats。

- 文本索引的表视图：
 - 事件视图
 - 日志表视图

数据库级别信息的视图

db2ext.dbdefaults 视图显示数据库的所有缺省值。

数据库级别上的缺省值不能更改，并且在此视图以“属性 — 值”对的形式提供：

db2ext.dbdefaults

db2 select DEFAULTNAME, DEFAULTVALUE from DB2EXT.DBDEFAULTS

表 8. db2ext.dbdefaults 视图

属性	缺省值	注意事项
CCSID	数据库的 CCSID	文档的缺省 CCSID。如果在 CREATE INDEX 命令中未指定任何 CCSID，则应用它。

表 8. db2ext.dbdefaults 视图 (续)

属性	缺省值	注意事项
FORMAT	TEXT	文档缺省格式。如果在 CREATE INDEX 命令中未指定任何格式，则应用它。
INDEXDIRECTORY	参见 注意事项 下面的路径名	全文本索引文件的目录。如果在 CREATE INDEX 命令中未指定任何索引目录，则应用它。路径名为: \$DB2EXT_INSTOWNERHOMEDIR/sql1lib/db2ext/indexes
LANGUAGE	EN_US	文档语言。
MODELCCSID	数据库的 CCSID	文档模型文件的 CCSID。
UPDATECOMMITCOUNT	0	更新期间在一个事务中处理的更改数。
CLEARCOMMITCOUNT	0	CLEAR INDEX 命令运行期间在一个事务中处理的更改数。
UPDATEFREQUENCY	NONE	何时检查新索引中的更新。
UPDATEMINIMUM	1	执行更新之前的最小更改数。
WORKDIRECTORY	参见 注意事项 下面的路径名	索引临时文件的目录。路径名为: <os_dependent>/sql1lib/db2ext/indexes
CACHEDIRECTORY	参见 注意事项 下面的路径名	CREATE INDEX 命令的 PERSISTENT CACHE 选项的缺省目录。路径名为: <os_dependent>/sql1lib/ db2ext/memory
PCTFREE	50	保留以供将来插入操作使用的高速缓存的百分比。
USERPERSISTENTCACHE	1	使用持久高速缓存。
AUTOMATICREORG	1	CREATE INDEX 命令中的 REORGANIZE 选项。这意味着自动重组。
TREATNUMBERSASWORDS	0	不要将序列解释为独立的词语，即使它们是相邻字符。例如，0 缺省值意味着 tea42at5 被视作一个词语。
INDEXSTOPWORDS	1	建立索引期间忽略无用词。
VERSION		NSE V8.1.2 NSE 的当前版本号。

表 8. *db2ext.dbdefaults* 视图 (续)

属性	缺省值	注意事项
UPDATEDELAY	0	只将日志条目用于带有时间戳记的增量更新。如果日志条目早于当前时间戳记，则表示出现更新延迟。这仅应用于更新命令运行期间对用户表长期运行的事务以避免丢失更新。注意，借助日志条目和增量更新，可以不使用任何捕获表。

db2ext.proxyinformation 表

如果您正在使用数据链路并且想要使用代理服务器存取文件，则必须在 *db2ext.proxyinformation* 表中指定代理信息。

表 9. *db2ext.proxyinformation* 视图

属性	类型	注意事项
PROXYHOST	VARCHAR(254)	代理服务器的主机名。
PROXYPORT	VARCHAR(6)	代理服务器的已使用端口。
PROXYTYPE	VARCHAR(10)	代理服务器的类型（PROXY 或 SOCKS）。
PROXYTIMEOUT	INTEGER	超时（以秒计）。

可以插入最多一行。如果在端口 123 上具有代理服务器，则可以使用以下 SQL 语句插入一行：

```
db2 insert into db2ext.proxyinformation values ('proxy1', '123', 'PROXY', 10)
```

索引级别信息的视图

可以使用下列 DB2 Net Search Extender 视图查询索引级别的信息：

- *db2ext.textindexes*
- *db2ext.textindexformats*
- *db2ext.indexconfiguration*
- `<index eventview name schema>.<index eventview name>`

由于向下兼容性的原因，DB2 Text Information Extender 视图 *db2ext.textcolumns*、*db2ext.formats* 和 *db2ext.models* 仍然受支持，但是不建议使用。

注意：在 *db2ext.textcolumns* 视图中，OPERATION、OPERATIONBEGIN 和 OPERATIONEND 列不再受支持。

db2ext.textindexes 视图

对 DB2 Net Search Extender 启用的每个数据库包含 db2ext.textindexes 视图。这包含有关在此数据库中创建的文本索引的设置、统计信息和缺省值的信息。

当创建文本索引时，会在 db2ext.textindexes 中创建新的条目。当删除文本索引时，会删除这些条目。

可以查询视图以获取有关索引的信息。以下是使用索引模式的示例：

```
db2 "select COLNAME from DB2EXT.TEXTINDEXES where INDSHEMA='myschema'
      and INDNAME='myindex'"
```

但要注意，不能使用常规 SQL 数据处理命令修改视图或显式创建或删除目录视图。视图的附加内容可在下表中找到。

还要注意，复制参数未包括在此视图中。

表 10. db2ext.textindexes 视图

属性	类型	注意事项
INDSCHEMA	VARCHAR(128)	文本索引的模式名。
INDNAME	VARCHAR(128)	文本索引的名称。
TABSCHEMA	VARCHAR(128)	基本表、昵称和视图的模式的表名。
TABNAME	VARCHAR(128)	对其创建索引的别名。
COLNAME	VARCHAR(128)	对其创建索引的列。
CCSID	INTEGER	此索引的文档 CCSID。
LANGUAGE	VARCHAR(5)	此索引的文档语言。
FUNCTIONSCHEMA	VARCHAR(128)	列映射函数的模式。
FUNCTIONNAME	VARCHAR(18)	列映射函数的名称。
INDEXDIRECTORY	VARCHAR(256)	全文本索引文件的目录。
WORKDIRECTORY	VARCHAR(256)	索引临时文件的目录。
CACHEDIRECTORY	VARCHAR(256)	持久高速缓存的目录（如果 persistentcache=1 的话）。
UPDATEFREQUENCY	VARCHAR(300)	将自动更新应用于此索引的触发器条件。
UPDATEMINIMUM	INTEGER	在更新执行之前必须更改的最小文档数。
EVENTVIEWSCHEMA	VARCHAR(128)	为此索引创建的事件视图的模式。
EVENTVIEWNAME	VARCHAR(128)	为此索引创建的事件视图的名称。
LOGVIEWSCHEMA	VARCHAR(128)	为索引创建的日志视图的模式。

表 10. db2ext.textindexes 视图 (续)

属性	类型	注意事项
LOGVIEWNAME	VARCHAR(128)	为索引创建的日志视图的名称（对视图的增量更新很重要）。
COMMITCOUNT	INTEGER	落实计数更新的缺省值。
NUMBER_DOCS	INTEGER	目前索引中的总文档数。注意，在索引更新期间，仅当设置 commitcount 时才更新此值。
REORG_SUGGESTED	INTEGER	指示通过运行 UPDATE INDEX REORGANIZE 是否能改进性能。仅当至少其中一个节点具有建议的索引重组时，此参数才为 true (1)。
REORGAUTOMATIC	INTEGER	1（如果更新操作期间索引自动重组的话）。
RECREATEONUPDATE	INTEGER	1（如果更新操作期间索引自动重组的话）。
CREATIONTIME	TIMESTAMP	索引创建的时间。
UPDATETIME	TIMESTAMP	上次更新的时间。如果 UPDATE TIME 等于 CREATION TIME，则尚未处理任何更新。
PERSISTENTCACHE	INTEGER	1（如果使用持久高速缓存的话）。
MAXIMUMCACHESIZE	INTEGER	高速缓存的最大大小。
PCTFREE	INTEGER	保留以供将来插入操作使用的高速缓存的百分比。
CACHETABLE	VARCHAR(32000)	CACHE TABLE 的列表表达式列表。
RESULTORDER	VARCHAR(32000)	INITIAL RESULT ORDER 的 SQL 排序依据。
ATTRIBUTES	VARCHAR(32000)	ATTRIBUTES 的列表表达式列表。
VIEWKEYCOLUMNS	VARCHAR(32000)	视图的索引的键列。

db2ext.indexconfiguration 视图

索引配置参数可从 db2ext.indexconfiguration 视图中获取。可通过常规 SQL 查询工具获取该视图。以下是使用索引名的示例：

```
db2 "select VALUE from DB2EXT.INDEXCONFIGURATION where INDSHEMA='myschema'
and INDNAME='myindex' and PARAMETER ='INDEXSTOPWORDS'"
```

视图的附加内容可在下列表中找到。

表 11. db2ext.indexconfiguration 视图

属性	类型	注意事项
INDSCHEMA	VARCHAR(128)	索引的模式名。
INDNAME	VARCHAR(128)	索引的名称。
PARAMETER	VARCHAR(30)	参数的类型。
VALUE	VARCHAR(512)	参数的值。

对于 PARAMETER 和 VALUE 属性，有几个值可用。

表 12. db2ext.indexconfiguration 视图

属性和值	属性和值
PARAMETER	VALUE
- TREATNUMBERASWORDS	- 0 或 1
- INDEXSTOPWORDS	- 0 或 1
- UPDATEDELAY	- 秒数大于等于 0

有关进一步信息，参见 CREATE INDEX 命令的 CONFIGURATION 选项。

db2ext.textindexformats 视图

可从 db2ext.textindexformats 视图中获取索引的格式和模型信息。以下是使用索引名的示例:

```
db2 "select FORMAT from DB2EXT.TEXTINDEXFORMATS where INDSCHEMA='myschema'
    and INDNAME='myindex'"
```

视图的附加内容可在下表中找到。

表 13. db2ext.textindexformats 视图

属性	类型	注意事项
INDSCHEMA	VARCHAR(128)	索引的模式名（用作日志表中的索引名和模式名的前缀）。
INDNAME	VARCHAR(128)	在 CREATE INDEX 命令中指定的索引名。
FORMAT	VARCHAR(30)	模型绑定至此格式。
MODELNAME	VARCHAR(30)	文档模型的名称。
MODELFILE	VARCHAR(256)	包含模型定义的文件。

表 13. *db2ext.textindexformats* 视图 (续)

属性	类型	注意事项
MODELCCSID	INTEGER	MODELFILE 的 CCSID。
DEFAULT	INTEGER	目前为 1 (原因是索引中的多个格式目前不受支持)。

文本索引的表视图

可以使用下列 DB2 Net Search Extender 视图来查询索引级别的信息:

- 事件视图
- 日志表视图

事件视图

此视图允许您获取有关建立索引状态、错误事件和建立索引期间何时发生问题 (例如, 找不到文档) 的信息。然后, 将这些索引更新事件写至索引的事件表。

模式和名称存储在 *db2ext.textindexes* 视图中。要获取事件视图的名称, 使用以下示例:

```
db2 "select EVENTVIEWSHEMA, EVENTVIEWNAME from DB2EXT.TEXTINDEXES
      where INDSHEMA = 'myschema' and INDNAME = 'myindex'
```

索引的事件视图包含下列各列。

表 14. 事件视图

属性	类型	注意事项
OPERATION	INTEGER	对用户表的操作反映在全文本索引 (insert = 0/ update = 1/ delete = 2) 中。
TIME	TIMESTAMP	创建事件条目的时间戳记。
REASON	INTEGER	原因代码。有关原因代码的列表, 参见第 263 页的附录 I, 『文本搜索引擎原因代码』。
SEVERITY	INTEGER	表条目的严重性。例如, 1 仅供参考, 4 指示警告, 而 8 意味着表条目错误。
MESSAGE	VARCHAR(1024)	附加文本信息。
KEY1, ... KEY14	依赖于用户表	用户表的第一个主键列至最后一个主键列 (最大为 14)。
PARTITION	INTEGER	在其上发生此错误的数据库分区号。在非分布式环境中, 这是 0。

可以使用 DB2TEXT CLEAR EVENTS 命令来清除事件，参见第 128 页的『CLEAR EVENTS』以获取进一步信息。

注意

还可在此视图中获取信息性事件（例如，启动、落实和完成更新处理）。

在这种情况下，Key1、... Key14 和 OPERATION 都具有空值。

对于视图的索引，PK01、...、PK14 列与在 CREATE INDEX 命令的 KEY COLUMNS 子句中指定的各列相关。

日志表、视图和昵称

日志表的用途就是存储对用户表或视图的更改操作，这些用户表或视图需要与外部全文本索引同步。

对于为常规表或昵称表创建的索引，有一些针对用户表创建的触发器将更改信息填入日志表中。但是，如果使用复制捕获表，则不创建任何日志表，反而使用复制捕获表。

对于日志表，更新命令读取各个条目，并在成功同步后删除它们。

但是，对于视图的索引，触发器不能填充日志表。因为可以更新视图，所以用户负责此任务。

表 15. 日志表视图

属性	类型	注意事项
OPERATION	INTEGER	对需要索引同步的用户表的更改的类型：（0 = 插入、1 = 更新，而 2 = 删除）。
TIME	TIMESTAMP	在此表中创建一行的时间戳记。
PK01 ... PKnm	与用户表相同	出现错误时发生问题的列。对于视图的索引，它们为用户表的主键列或同等键列的副本。

创建该表的用户能够选择、更新、插入和删除此视图。

如果在创建索引命令中指定复制捕获表，则不创建任何日志表反而使用复制捕获表。复制捕获表必须包含下列各列：

表 16. 复制捕获表

属性	类型	注意事项
IBMSNAP_OPERATION	INTEGER	对需要索引同步的 CD 或 CCD 表的更改的类型：（I = 插入、U= 更新，而 D= 删除）。
IBMSNAP_COMMITSEQ	CHAR	映射至 CD 或 CCD 表的相应列。
IBMSNAP_INTENTSEQ	CHAR	映射至 CD 或 CCD 表的相应列。
PK01 ... PKnm	与用户表相同	出现错误时发生问题的列。它们是用户表的主键列。

定义该表的用户能够使用授权选项进行选择、更新、插入和删除。

附录 D. 受支持的 CCSID

DB2 Net Search Extender 支持下列 CCSID。

CCSID

37	美国 / 加拿大 — CECP
273	德国 / 奥地利 — CECP
274	旧的比利时代码页
277	丹麦和挪威 — CECP
278	芬兰和瑞典 — CECP
280	意大利 — CECP
284	西班牙 / 拉丁美洲 — CECP
285	英国 — CECP
290	扩展日语（片假名）
297	法国 — CECP
301	日本 DB PC
367	ASCII
420	阿拉伯双语
423	希腊 — 183
424	以色列（希伯来语）
437	美国英语
500	世界语 #5
737	MS DOS 希腊语
806	印地语
813	希腊语
819	拉丁语-1
833	扩展韩国语
836	扩展简体中文

838	使用了低调标记和古代字符的泰国语
848	使用了欧元符号的 PC、西里尔语和乌克兰语
850	拉丁语-1
852	拉丁语-2
855	保加利亚语
857	土耳其语
858	个人计算机 — 使用了欧元符号的多种语言
860	葡萄牙语
862	希伯来语
863	加拿大语
864	阿拉伯语
866	俄语
867	以色列 — 个人计算机
869	希腊语
870	拉丁语 2 — EBCDIC 多种语言
871	冰岛 — CECP
872	西里尔语 — 使用了欧元符号的 PC
874	泰国语
875	希腊
891	韩国 — 个人计算机
895	日本 7 位拉丁语
901	使用了欧元符号的 PC 波罗的海多种语言
902	使用了欧元符号的 8 位爱沙尼亚语
912	拉丁语-2
915	俄语
916	希伯来语
920	土耳其语
921	拉脱维亚语, 爱沙尼亚语
922	爱沙尼亚语

923	拉丁语 9
924	拉丁语 9 EBCDIC
927	台湾 PC
930	日本 EBCDIC
932	日语, 组合的 SBCS/DBCS
933	韩国语
935	中文 (简体)
937	中文 (繁体)
938	台湾 PC
939	日语
941	日本 OPEN
942	日语, 组合的 SBCS/DBCS
943	日语, 组合的 SBCS/DBCS
948	中文 (繁体), 组合的 SBCS/DBCS
949	韩国语
950	中文 (繁体), 组合的 SBCS/DBCS
954	日语
964	中文 (繁体), 组合的 SBCS/DBCS
970	韩国语
1025	西里尔语 (多种语言)
1026	拉丁语 #5 — 土耳其
1027	扩展的日语 (拉丁语)
1040	扩展的韩国语 — 个人计算机
1041	扩展的日语 — 个人计算机
1043	扩展的繁体中文 — PC
1046	阿拉伯语
1047	拉丁语 1 / 开放式系统
1051	H-P 仿真, 罗马数字 8
1088	已修订的韩国语 — 个人计算机

1089	阿拉伯语
1112	波罗的海 — 多种语言, EBCDIC
1115	中华人民共和国 (PRC) — PC
1122	爱沙尼亚, EBCDIC
1123	西里尔 / 乌克兰
1124	乌克兰语
1125	乌克兰语
1131	越南语
1137	天城书 EBCDIC
1140	美国和加拿大等 ECECP
1141	奥地利和德国 ECECP
1142	丹麦和挪威 ECECP
1143	芬兰和瑞典 ECECP
1144	意大利 ECECP
1145	西班牙和拉丁美洲 (西班牙语) ECECP
1146	英国 ECECP
1147	法国 ECECP
1148	世界语 ECECP
1149	冰岛 ECECP
1153	使用了欧元符号的 EBCDIC 拉丁语 2 多种语言
1154	使用了欧元符号的 EBCDIC 西里尔多种语言
1155	使用了欧元符号的 EBCDIC 土耳其语
1156	使用了欧元符号的 EBCDIC 波罗的海多种语言
1157	使用了欧元符号的 EBCDIC 爱沙尼亚语
1158	使用了欧元符号的 EBCDIC 西里尔语 / 乌克兰语
1160	使用了低调标记和古代字符的泰国语
1161	使用了低调标记和古代字符的泰国语 — PC
1162	泰国语 MS Windows
1163	使用了欧元符号的越南语 ISO-8

1164	使用了欧元符号的越南语 EBCDIC
1200	UCS2
1208	UTF8
1250	拉丁语-2, 白俄罗斯语
1251	俄语
1252	拉丁语-1
1253	捷克语
1254	土耳其语
1255	希伯来语
1256	阿拉伯语
1258	越南语
1275	Apple, 拉丁语 1
1280	Apple 希腊语
1281	Apple 土耳其语
1282	Apple 中欧
1283	Apple 西里尔语
1351	日本 OPEN
1363	韩国语
1364	韩国语
1381	中文(简体), 组合的 SBCS/DBCS
1383	中文(简体), 组合的 SBCS/DBCS
1386	中文(简体), 组合的 SBCS/DBCS
1388	中文(简体), 组合的 SBCS/DBCS
1390	日语
1392	中国 GB18030
1394	
1399	日本 EBCDIC
4909	希腊 / 拉丁语 ASCII
4930	韩国 DB EBCDIC

4933	中国 EBCDIC
4971	希腊 EBCDIC
5026	日语片假名
5035	日语拉丁语
5039	日语, 组合的 SBCS/DBCS
5210	中国 SB PC
5346	Windows 拉丁语-2
5347	Windows 西里尔语
5348	Windows 拉丁语-1
5349	Windows 希腊
5350	Windows 土耳其
5351	使用了欧元符号的 Windows 希伯来语
5352	Windows 阿拉伯语
5353	Windows 波罗的海语
5354	越南语
9044	拉丁语-2 PC
9048	希伯来语 PC
9049	土耳其 PC
9061	希腊 PC
9238	阿拉伯语 — PC
12712	希伯来语 EBCDIC
13121	韩国 SB EBCDIC
13488	UCS2
17248	阿拉伯语 PC
17584	UCS-2
18030	
21427	台湾 BIG-5
33722	IBMecJP
61955	

65500
65501

附录 E. 受支持的语言

有一些语言参数，可以在 DB2 Net Search Extender 中指定它们。在 Net Search Extender 中，唯一有效的特定于语言的处理是针对用泰国语编写的文档完成的（或者如果请求了的话，则是在无用词处理期间完成的）。有关更多信息，参见第 261 页的『无用词』。

AR_AA 阿拉伯语 / 阿拉伯国家或地区使用

BE_BY 白俄罗斯语 / 白俄罗斯

BG_BG 保加利亚语 / 保加利亚

CA_ES 加泰罗尼亚语 / 西班牙

CS_CZ 捷克语 / 捷克共和国

DA_DK 丹麦语 / 丹麦

DE_CH 德语 / 瑞士

DE_DE 德语 / 德国

EL_GR 希腊语 / 希腊

EN_AU 英语 / 澳大利亚

EN_BE 英语 / 比利时

EN_GB 英语 / 英国

EN_US 英语 / 美国

EN_ZA 英语 / 南非

ES_ES 西班牙语 / 西班牙

ET_EE 爱沙尼亚语 / 爱沙尼亚

FI_FI 芬兰语 / 芬兰

FR_BE 法语 / 比利时

FR_CA 法语 / 加拿大

FR_CH 法语 / 瑞士

FR_FR 法语 / 法国

HE_IL 希伯来语 / 以色列

HI_IN	印地语 / 印度
HR_HR	克罗地亚语 / 克罗地亚
HU_HU	匈牙利语 / 匈牙利
ID_ID	印度尼西亚语 / 印度尼西亚
IT_CH	意大利语 / 瑞士
IW_IL	希伯来语 / 以色列
IT_IT	意大利语 / 意大利
JA_JP	日语 / 日本
KO_KR	韩国语 / 韩国
LT_LT	立陶宛语 / 立陶宛
LV_LV	拉脱维亚语 / 拉脱维亚
MK_MK	马其顿语 / FYR 马其顿
MS_MY	马来语 / 马来西亚
NB_NO	(挪威) 博克马尔语 / 挪威
NL_BE	荷兰语 / 比利时
NL_NL	荷兰语 / 荷兰
NN_NO	(挪威) 尼诺斯克语 / 挪威
NO_NO	挪威语 / 挪威
PT_BR	葡萄牙语 / 巴西
PL_PL	波兰语 / 波兰
PT_PT	葡萄牙语 / 葡萄牙
RO_RO	罗马语 / 罗马
RU_RU	俄语 / 俄罗斯
SH_SP	塞尔维亚语 (拉丁语) / 塞尔维亚
SK_SK	斯洛伐克语 / 斯洛伐克
SL_SI	斯洛文尼亚语 / 斯洛文尼亚
SQ_AL	阿尔巴尼亚语 / 阿尔巴尼亚
SR_SP	塞尔维亚语 (西里尔语) / 塞尔维亚
SV_SE	瑞典语 / 瑞典

TA_IN	泰米尔语 / 印度
TE_IN	泰卢固语 / 印度
TH_TH	泰国语 / 泰国
TR_TR	土耳其语 / 土耳其
UK_UA	乌克兰语 / 乌克兰
VI_VN	越南语 / 越南
ZH_CN	中文 / 中华人民共和国
ZH_TW	中文 / 台湾

附录 F. Net Search Extender 消息

DB2 Net Search Extender 提供下列消息类型:

- 信息和警告消息
- 错误消息

注意, 从搜索函数返回的 SQL 状态是 38600 和 CTE 错误号。

信息和警告消息

CTE0001	操作成功完成。
CTE0002	更新和锁定服务已启动且正在运行。
CTE0003	索引更新已开始。
CTE0004	索引更新已结束。
CTE0005	索引更新落实: 成功插入、更新和 / 或删除了 “%1”、 “%2” 和 “%3” 文档。
CTE0006	访问文本索引时出现问题。有关详细信息, 检查 db2diag.log。
CTE0007	节 “%1” 没有出现在这些文档的任何一个中, 或它是无效文档模型节名。
CTE0008	索引重组已开始。
CTE0009	索引重组已结束。
CTE0010	属性 “%1” 无效。
CTE0011	高速缓存激活已开始。
CTE0012	高速缓存激活已结束。
CTE0013	持久高速缓存已除去。
CTE0014	高速缓存已取消激活。

错误消息

CTE0100 **DB2 操作失败。DB2 信息**
 “%2” “%4”。

解释: 发生了 DB2 错误，不允许进一步处理。

措施: 有关此 DB2 错误的更详细的信息，使用 db2 ? SQLxxx。

CTE0101 **搜索引擎操作失败。原因代码:**
 “%2”、“%3”、“%4”、“%5”
 和“%6”。

解释: 发生了“搜索引擎”错误，不允许进一步处理。

措施: 有关更详细的信息，参见“搜索引擎”原因代码描述。

CTE0102 **常规系统函数失败。错误: “%2”。**

解释: 发生了系统错误，不允许进一步处理。

措施: 可以在 UNIX 上的 errno.h 头文件找到附加信息。

CTE0103 **发生了内部错误。位置: “%1”和**
 “%2”。

解释: 发生了内部处理错误，不允许进一步处理。尝试启动并停止更新和锁定服务以及 DB2。

措施: 如果仍然存在错误，则启动跟踪并检查 db2diag.log。

CTE0104 **内存分配错误（搜索引擎）。**

解释: 系统内存不足。

措施: 增加实例所有者的可用内存大小或停止并行运行的其它进程。

CTE0105 **内存分配错误。**

解释: 系统内存不足。

措施: 增加用户的可用内存大小或停止并行运行的其它进程。

CTE0106 **表“%1”。“%2”没有任何主键。**

解释: 已尝试对没有主键的表创建索引。

措施: 调用 db2 alter table 以确保主键存在。然后再次尝试创建索引。

CTE0107 **目录“%1”不存在。**

解释: 您指定了不存在的目录。

措施: 创建目录，并确保实例所有者可以存取它。然后再次尝试指定目录。注意，在分布式 DB2 环境中，每个物理节点上都必须存在此目录。

CTE0108 **对象“%1”。“%2”的键列的内部**
 大小“%4”比允许的最大大小
 “%3”还要大。

解释: 键列的内部表示法超出最大大小。

措施: 再次创建索引前更改表的布局。使用较小的键列，这还可以提高性能。

CTE0109 **对象“%1”。“%2”的键列“%3”**
 数比允许的最大值“%4”还要大。

解释: 最多支持 14 个键列。

措施: 再次创建索引之前更改表的布局。

CTE0111 **文件“%1”不可读。**

解释: 不能读取指定的文件。

措施: 检查文件的访问权。考虑将“存储过程”作为受防护的用户标识来运行，这可能还需要使用此文件的权限。

CTE0112 **打不开文件“%1”。**

解释: 打不开指定的文件。

措施: 验证是否正确地指定了文件。

CTE0113 将模型文件 “%1” 转换为 UTF8 编码时出错。

解释: 指定的 CCSID 或缺省数据库 CCSID 与模型文件 CCSID 不匹配。

措施: 确保正确指定模型文件 CCSID。

CTE0114 无法在文件 “%2” 中注册文档模型 “%1”。

解释: 未能使用模型文件。

措施: 检查是否正确地指定了模型文件语法。

CTE0115 发生了锁定问题。锁定管理器信息: “%1” “%2”。

解释: 发生了内部锁定问题。

措施: 使用 db2text control 命令来检查当前锁定。使用同一命令, 清除悬挂锁定。如果这样做没什么用, 则停止并重新启动锁定和更新服务。

CTE0116 操作与现有锁定冲突。

解释: 已尝试在对索引运行其它命令时使用当前不允许的命令。

措施: 检查对此索引持有的锁定以了解当前正在运行哪些命令。等待直到其它命令完成。如果操作不再运行而锁定仍然活动, 则清除索引的锁定并重试。

CTE0117 已使用数据库的所有可用锁定空间。更改配置。

解释: 已尝试使用超过锁定文件中配置的数目的数据库。

措施: 在锁定配置 db2extlm.cfg 中更改想要并行使用的数据库的数目。使用 db2text stop 和 db2text start 命令重新启动更新和锁定服务。

CTE0118 已使用数据库的索引的所有可用锁定空间。更改配置。

解释: 已尝试对一个数据库使用超过在锁定文件中配置的数目的索引。

措施: 在锁定配置文件 db2extlm.cfg 中更改想要并行使用的索引的数目。使用 db2text stop 和 db2text start 命令重新启动更新和锁定服务。

CTE0119 已使用索引的锁定的所有可用空间。

解释: 正在运行的操作针对一个索引需要的锁定比锁定配置文件中配置的还要多。

措施: 在锁定配置 db2extlm.cfg 中更改想要并行使用的锁定的数目。使用 db2text stop 和 db2text start 命令重新启动更新和锁定服务。

CTE0120 更新和锁定服务配置文件错误。

解释: 配置文件 db2extlm.cfg 出错。

措施: 检查 db2extlm.cfg 文件并更正错误。使用 db2text start 命令重新启动更新和锁定服务。

CTE0121 打不开更新和锁定服务配置文件。

解释: 打不开 db2extlm.cfg 文件。

措施: 检查文件是否存在以及是否可存取它。如果文件不能存取, 尝试使用 db2iupdt 更新 db2 实例。

CTE0122 在更新和锁定服务配置文件中发现语法错误。

解释: 在更新和锁定服务配置文件中发现语法错误。

措施: 检查更新和锁定服务配置文件以查找错误。

CTE0126 更新和锁定服务输入文件 “%1” 已损坏。

解释: 更新和锁定服务的必需文件已损坏。

措施: 检查文件是否存在以及是否可存取它。如果

可存取该文件，则重命名该文件并重新启动更新和锁定服务。应该再次创建该文件。但是，此操作将除去创建索引的所有指定频率更新。

CTE0127 发生了更新和锁定服务错误。原因代码: “%1”。

解释: 在更新和锁定服务区域中发生了内部错误。
措施: 停止 DB2 和 Net Search Extender，然后清除共享资源。再次尝试启动这两项。如果这样做不起作用，则向 IBM 代表报告该问题。

CTE0129 不允许将空值作为参数传递。

解释: DB2 已将空值传递给内部用户定义函数。
措施: 首先确保指定基本表具有主键。更改选择语句以避免此问题。打开跟踪功能，并将返回的信息传递给 IBM 服务中心。

CTE0130 指定的搜索自变量超出最大长度。当前搜索自变量长度为 “%1”，而最大受支持长度为 “%2”。

解释: 指定的搜索自变量的长度为 “%1”。最大长度一定不能超过 “%2”。
措施: 将搜索自变量的长度减少至 “%2”。

CTE0131 用户定义函数 “%1” . “%2” 不存在。

解释: 指定的用户定义函数在此数据库中不存在。
措施: 检查为此用户定义函数指定的名称，或在正在使用的数据库中注册该用户定义函数。

CTE0132 文本索引 “%1” . “%2” 不存在。

解释: 指定的文本索引在此数据库中不存在。
措施: 检查指定的名称和正在使用的数据库。使用 db2ext.textcolumns 视图来查看现有文本索引。

CTE0133 文本索引 “%1” . “%2” 已存在。

解释: 指定的文本索引在此数据库中已存在。
措施: 检查指定的名称和正在使用的数据库。使用 db2ext.textcolumns 视图来查看现有文本索引。

CTE0135 对象 “%1” . “%2” 不存在。

解释: 指定的对象名在此数据库中不存在。
措施: 检查指定的对象名和正在使用的数据库。

CTE0136 列 “%1” 在 “%2” . “%3” 中不存在。

解释: 指定的列不存在。
措施: 检查指定的列名。检查正在使用的表、视图或数据库。

CTE0137 表空间 “%1” 不存在。

解释: 指定的表空间在此数据库中不存在。
措施: 检查指定的名称和正在使用的数据库。

CTE0138 表空间 “%1” 不是常规表空间。

解释: 指定的表空间不是常规表空间。只能在常规表空间中创建事件表。
措施: 再次将此命令与常规表空间配合使用。

CTE0139 未设置环境变量 “%1”。

解释: 未设置必需的环境变量。
措施: 检查环境，指定必需的变量，并再次使用该命令。

CTE0140 已启用数据库 “%1” 的文本搜索。

解释: 已启用指定数据库的文本搜索。
措施: 检查指定的名称。还要检查隐含隐式连接的 DB2DBDFT 变量。

CTE0141 未启用数据库 “%1” 的文本搜索。

解释: 未启用指定数据库的文本搜索。

措施: 检查指定的数据库名和 DB2DBDFT 变量。如果数据库名正确, 则使用命令 `db2text enable database for text`。

CTE0142 命令需要授予用户 “%3” 对 “%1” . “%2” 的控制权限。

解释: 您没有使用此命令的权限。

措施: 只有此表的所有者才能使用此命令或为您提供必需的权限。

CTE0143 对于用户 “%1”, 命令需要数据库管理权限。

解释: 您没有使用此命令的必需权限。

措施: 只有数据库的所有者才能使用此命令或为您提供必需的权限。

CTE0144 在数据库 “%1” 中至少有一个文本索引是活动的。

解释: 在删除所有文本索引之前, 不能禁用数据库。

措施: 参见 `db2ext.textcolumns` 视图以获取现有索引。使用 `DROP INDEX` 命令删除现有索引或使用 `DISABLE DATABASE` 命令指定 `FORCE` 选项。

CTE0145 CCSID “%1” 不受支持。

解释: 指定的 CCSID 不受支持。

措施: 指定了无效的 CCSID。

CTE0146 语言 “%1” 不受支持。

解释: 指定的语言不受支持。

措施: 指定了无效的语言。

CTE0147 格式 “%1” 不受支持。

解释: 指定的格式不受支持。

措施: 指定了无效的格式。

CTE0148 指定的格式 “%1” 不接受模型文件。

解释: 格式 “%1” 不支持模型文件。

措施: 使用接受模型文件的格式或从命令中除去模型文件。

CTE0149 对索引更新频率指定的术语过多 (从 “%1” 开始)。

解释: 更新频率的语法不正确。

措施: 确保只指定 `DAY`、`HOUR` 和 `MINUTE` 参数一次。

CTE0150 命令意外结束。检查命令语法。

解释: 命令语法不正确。

措施: 检查命令语法。验证是否指定了必需的参数。

CTE0151 标记 “%1” 意外出现。检查命令语法。

解释: 命令的语法不正确。

措施: 检查命令语法, 并验证正在使用的标记在特定命令中是否允许使用。

CTE0152 标记 “%1” 过长。

解释: 标记过长。

措施: 检查命令语法, 并验证标记是否减少到允许的最大大小。

CTE0153 标记“%1”在更新频率中出现两次。

解释: 对更新频率指定了不正确的语法。

措施: 确保只指定 DAY、HOUR 和 MINUTE 参数一次。

CTE0154 “%2”的值“%1”超出范围。有效范围是“%3”至“%4”。

解释: 指定了不正确的值。值应该在允许的范围

措施: 更新命令。更改该值以使它与允许范围内的值相匹配。

CTE0155 搜索字符串是空的。

解释: 指定了空的搜索字符串。

措施: 检查搜索字符串是否包括有效的字母数字字符。

CTE0157 “%1”附近存在语法错误。

解释: 指定了不正确的搜索语法。

措施: 检查 %1 附近的语法。更正错误并重试。

CTE0158 缺少自由文本搜索字符串。

解释: 指定自由文本字符串。

措施: 检查“is about”后面的搜索字符串是否包括有效字母数字字符。

CTE0159 搜索字符串超出允许的长度“%1”。

解释: 搜索字符串过长。

措施: 减少搜索字符串的大小并重试。

CTE0160 尚未在搜索字符串中指定任何节名。

解释: 需要指定有效节名。

措施: 添加有效节名并重试。

CTE0162 未能处理转义命令。

解释: 搜索字符串包括可用作屏蔽字符的特殊字符过多。

措施: 减少搜索术语中的特殊字符数，或避免使用转义命令。可以使用下列特殊字符: ! * + , _ . : ; { } ~ | ? [] ` = \

CTE0163 在同义字子句中未指定任何同义字名称。

解释: 请求了同义字搜索而未使用同义字名称。

措施: 在搜索自变量中指定同义字名称。

CTE0164 同义字关系“%1”中存在语法错误。

解释: 同义字关系的指定语法不正确。

措施: 按照语法规则更新同义字关系。

CTE0166 自由文本一定要是搜索查询中的最后一个语句。

解释: “is about”标记后面不允许再有运算符。

措施: 重写查询字符串。最后一个运算符一定要是“is about”。

CTE0167 自由文本查询“%1”中存在语法错误。

解释: 自由文本字符串的语法不正确。

措施: 按照语法规则更新自由文本字符串。

CTE0168 节语句中缺少左括号。

解释: 节语句的语法不正确。

措施: 按照语法规则更新节语句。

CTE0169 节语句中缺少逗号或右括号。

解释: 节语句的语法不正确。

措施: 按照语法规则更新节语句。

CTE0170	缺少右双引号。
解释:	搜索术语的指定语法不正确。
措施:	按照语法规则更新搜索术语。
CTE0171	缺少节名的左双引号。
解释:	节语句的语法不正确。
措施:	按照语法规则更新节语句。
CTE0172	缺少节名的右双引号。
解释:	节语句的语法不正确。
措施:	按照语法规则更新节语句。
CTE0173	必须在转义子句中定义一个转义字符。
解释:	转义子句中只能存在一个字符。
措施:	除去转义子句中的附加字符。
CTE0174	不允许将空白字符作为转义字符。
解释:	在转义子句中不允许有空白字符。
措施:	将转义子句更改为具有有效字符的子句。
CTE0175	定义了转义子句但在搜索短语中找不到任何屏蔽字符。
解释:	在未使用屏蔽字符的情况下指定了转义子句。
措施:	除去转义子句。
CTE0176	短语中的转义字符的后续字符既不是同一字符也不是屏蔽字符。
解释:	转义字符后面的字符必须是屏蔽字符或转义字符本身。
措施:	更改搜索字符串以正确使用转义字符。

CTE0177	编号值“%1”无效。
解释:	搜索自变量中的指定编号无效。
措施:	检查文档以了解有效范围。更新搜索自变量中的值。
CTE0178	模糊短语中的屏蔽字符前必须加上转义字符。
解释:	不允许屏蔽与模糊搜索一起使用。
措施:	使用转义字符更新搜索字符串。
CTE0179	同义字名称“%1”超出允许的长度“%2”。
解释:	不支持长度超过 60 个字节的主键。
措施:	再次创建索引之前更改表的布局。
CTE0180	找不到同义字“%1”。
解释:	找不到指定的同义字。
措施:	检查同义字文件是否位于同义字目录中，或者它是否是全限定的。
CTE0181	不能装入库“%1”。
解释:	找不到库。
措施:	检查库是否位于库路径中以及是否可用。启动并停止 DB2 以确保使用当前设置。
CTE0182	不能从库“%2”装入函数“%1”。
解释:	不能装入库入口点。
措施:	存取的库可能是无效的。检查该库是否只指定了一次。
CTE0183	使用共享系统资源时发生了错误。
解释:	不能实现对共享系统资源（例如，共享内存或信号量）的请求。
措施:	检查当前系统状态和配置。在 UNIX 上，使用 ipcs 命令来检查资源。停止所有应用程序（例

错误消息

如, DB2 和 DB2 Net Search Extender)。如果列示了更多资源, 则使用 ipcrm 来清除它们。

CTE0184 未发出任何 db2text start 命令。

解释: 调用了需要锁定和更新服务的命令。

措施: 使用 db2text start 启动更新和锁定服务。

CTE0185 更新和锁定服务已经是活动的。

解释: 发出了 db2text start, 但是更新和锁定服务已在运行。

措施: 没有任何进一步的操作。

CTE0186 发生了更新和锁定服务错误, 有关详细信息, 请检查 db2diag.log。

解释: 发生了更新和锁定服务错误。

措施: 检查 db2diag.log 以了解进一步的信息, 或者清除共享资源。另见 CTE0183。

CTE0187 更新和锁定服务仍然是活动的, 使用 FORCE 选项来停止这些服务。

解释: db2text stop 命令尚未停止锁定服务, 仍然有进程正在运行。

措施: 使用 db2text control 检查哪些进程正在运行并等待这些进程完成。如果需要停止它们, 则使用 FORCE 选项。

CTE0188 使用更新和锁定服务时出现临时间题。请重试。

解释: db2text stop 命令尚未停止锁定服务。程序仍然在运行或发现不一致状态。

措施: 使用 db2text control 检查哪些进程正在运行并等待这些进程完成。要停止它们, 使用 FORCE 选项。

CTE0189 找不到可执行程序 “%1”。

解释: 找不到程序文件或不能存取它。

措施: 检查程序文件是否位于 DB2 服务器的 bin 或 adm 目录中。如果找不到该文件, 则安装会损坏。

CTE0190 不能启动可执行程序 “%1”。

解释: 不能启动程序。

措施: 检查程序是否位于 DB2 服务器的 bin 或 adm 目录中并且安装了适当的库。有关进一步信息, 在服务器上手工调用该程序。

CTE0191 删除索引操作未完成。有关详细信息, 检查 db2diag.log。

解释: 删除索引操作未完成, 可能是由 FORCE 选项导致的。

措施: 使用 FORCE 选项会删除所有内容而不考虑任何错误。检查索引目录以找出暂挂文件并手工除去这些文件。

CTE0192 更新索引操作中发生了错误。有关详细信息, 检查事件表 “%1” . “%2” 和 db2diag.log。

解释: 在索引更新过程期间, 所有文档错误都会写至事件表。

措施: 有关文档错误的更多信息, 检查事件表。在修正问题之后清除事件日志。

CTE0194 列 “%2” 的类型 “%1” 不受支持。

解释: 使用的列不在受支持列的列表中。

措施: 检查创建索引以获取 “键和建立索引” 的有效列的列表。对命令作适当更改并重试。

CTE0195 “%1” 不是绝对路径。

解释: 服务器上的绝对路径是必需的。

措施: 检查路径并在命令中填写绝对路径。

CTE0198 没有任何相应的文本索引。

解释: 列没有任何文本索引。

措施: 检查文本索引是否仍然存在。

CTE0199 没有与表 “%2” 的列 “%1” 相对应的文本索引。

解释: 已尝试搜索没有文本索引的列。

措施: 检查正在搜索的列或对该列创建文本索引。

CTE0200 必须指定至少一个命令选项。

解释: ALTER INDEX 命令更改索引的特征（例如，更新和存储器选项）。尚未指定要更改的任何特征。

措施: 指定至少一个命令选项。有关所有可能的选项，参考命令语法。

CTE0201 存在与同一列上的现有文本索引的冲突。

解释: 对同一列定义的文本索引是使用此创建索引命令中的不同参数创建的。

措施: 在创建索引命令中更正参数值。确保下列参数对现有索引和要创建的索引具有相同值：ccsid、语言、格式、文档模型、索引配置、列函数和属性。

CTE0202 指定键列时，对象 “%1” . “%2” 一定要是视图。

解释: 指定的对象不是视图。对视图的列建立索引时，只允许使用 KEY COLUMNS FOR INDEX ON VIEW 子句。

措施: 除去 KEY COLUMNS FOR INDEX ON VIEW(SQL-columnname-list) 子句。

CTE0203 未使用 **CACHE TABLE** 选项创建文本索引 “%1” . “%2”。这是执行命令所必需的。

解释: 只有使用 CACHE TABLE 选项创建指定的索引，才能执行此命令。

措施: 使用 CACHE TABLE 选项创建索引。有关命令语法，参考文档。

CTE0204 缺少属性名。将 “AS <attribute name>” 添加至属性表达式。

解释: 每当在属性表达式中使用列表表达式时，都必须提供属性名。例如：（C1+C2 AS myname）。

措施: 将 “AS <attribute name>” 添加至属性表达式。

CTE0205 **CACHE TABLE** 表达式无效。

解释: 高速缓存表表达式中的列列表无效。

措施: 更正创建索引命令中的高速缓存表列列表。确保指定的表中存在这些列。如果对列应用了函数，则验证是否正确地使用了函数。

CTE0206 **ATTRIBUTE** 表达式无效。

解释: 属性表达式中的列列表无效。

措施: 更正创建索引命令中的属性列列表。确保指定的表中存在这些列。如果对列应用了函数，则验证是否正确地使用了函数。

CTE0207 未对视图 “%1” . “%2” 的索引指定 **KEY COLUMNS FOR INDEX ON VIEW**。

解释: 如果创建了视图的索引，则必须指定 KEY COLUMNS FOR INDEX ON VIEW(SQL-columnname-list) 子句。列名称列表指定“唯一地”标识视图中的某行的各列。

措施: 将 KEY COLUMNS FOR INDEX ON VIEW(SQL-columnname-list) 子句包括在创建索引命令中。

**CTE0208 INITIAL SEARCH RESULT
ORDER 列无效。**

解释: INITIAL SEARCH RESULT
ORDER(SQL-order-by list) 表达式中的列列表无效。

措施: 更正创建索引命令中的排序列列表。检查语法是否正确以及指定的表中是否存在这些列。如果对列应用了函数，则验证是否正确地使用了函数。

CTE0209 属性列 “%2” 的类型 “%1” 不受支持，类型 DOUBLE 是必需的。

解释: 对于属性列，唯一受支持的数据类型是 DOUBLE。

措施: 确保具有要建立索引的文本列的表的属性列类型为 DOUBLE。可在属性表达式中使用转型运算符。有关可转型为双精度的数据类型，参考 SQL Reference。

CTE0210 索引配置参数 “%2” 的值 “%1” 无效。有效值为 “%3”。

解释: 配置参数的指定值不正确。有关参数的有效值，参考命令语法。

措施: 更正创建索引命令中的索引配置参数值。

CTE0211 “%1” 是无效的索引配置参数。

解释: 索引配置选项未知。

措施: 检查创建索引命令语法。有效的索引配置选项包括 TreatNumbersAsWords 和 IndexStopWords。下列各项必须用逗号隔开：index configuration(treatnumberaswords 1, indexstopwords 1)。

CTE0212 未能保存内部索引配置文件 “%1”。

解释: 未能保存索引的内部配置文件。

措施: 确保实例所有者对应保存文件的目录具有写许可权。如果同名文件已存在，则确保它对于实例所有者是可写的。

CTE0213 未能装入内部索引配置文件模板 “%1”。

解释: 未能读取内部索引配置文件模板。

措施: 确保文件存在于正确的位置，并且是可读的。

CTE0214 对索引配置文件设置新条目 “[%1],%2=%3” 时发生内部错误。

解释: 编写索引的内部配置文件时发生内部错误。

措施: 如果文件存在，则检查对于实例所有者它是否为可读 / 写的。检查文件所在的设备上是否有足够的空间。

CTE0215 对别名 “%1” . “%2” 创建索引不受支持。使用基本表 “%3” . “%4” 来代替。

解释: 不能对别名创建索引。

措施: 输入带有基本表的创建索引命令。

CTE0217 调度服务已经是活动的。

解释: 服务已经是活动的，不需要启动它。

措施: 不需要任何操作。

CTE0218 函数 “%1” 失败，错误代码为 “%2”。

解释: Windows 函数失败，出现指定错误代码，不允许进一步处理。

措施: 使用指定的 Windows 系统错误代码来获取详细的错误信息。

CTE0219 打不开服务 “%1”。错误代码为 “%2”。

解释: 在 Windows 系统上找不到指定的服务。

措施: 检查在 Windows 系统上是否安装了指定服务。使用指定的 Windows 系统错误代码来获取详细的错误信息。

CTE0220 找不到 DB2 实例概要文件路径。

解释: 用来获取 DB2 实例概要文件路径的内部 DB2 函数失败。

措施: 创建 DB2 实例而不指定实例概要文件路径信息并重试命令。

CTE0221 未正确指定 UpdateFrequency “%1”。

解释: 更新频率语句的语法不正确。

措施: 按照语法规则更正更新频率语句。

CTE0222 调度服务输入文件 “%1” 已损坏。

解释: 包含索引更新信息的调度程序文件已损坏。

措施: 使用系统编辑器并尝试更正问题。可能已截断条目，或者已删除结束行字符。如果这样做不能复原文件内容，则尝试以下操作：

- 调用命令 db2text stop 来停止调度程序。
- 删除调度程序服务文件。
- 调用 db2text start 命令来启动调度程序。
- 使用 db2text alter index ... 命令来重新创建所有相关索引的更新频率条目。

CTE0223 未能关闭文件 “%1”。

解释: 不能关闭指定的文件。

措施: 验证是否正确地指定了文件。

CTE0224 未能将文件 “%1” 复制至 “%2”。

解释: 不能将第一个文件复制至第二个文件。

措施: 验证是否正确地指定了文件。检查第二个文件是否已存在并且是只读的。还应检查系统上是否有足够的可用空间。

CTE0225 未能除去文件 “%1”。

解释: 不能从系统除去指定的文件。

措施: 验证是否正确地指定了文件并检查文件存取权限。

CTE0225 未能除去文件 “%1”。

解释: 不能从系统除去指定的文件。

措施: 验证是否正确地指定了文件并检查文件存取权限。

CTE0227 对文件 “%1” 的写操作失败。

解释: 指定的文件不是可写的。

措施: 验证是否正确地指定了文件并检查文件文件存取权限。还应检查系统上是否有足够的可用空间。

CTE0228 用户在操作系统级别没有足够的存取权限。

解释: 命令需要操作系统级别的管理员权限。

措施: 确保您具有操作系统管理员权限。检查您是否是管理员组的成员。

CTE0231 未在表空间 “%2” . “%3” 所在的节点组 (“%4”) 中定义 “%1”。

解释: 管理表的表空间需要分布在不同的节点之间，分布方式与包含要建立索引的文本列的表的分布方式完全相同。为了强制执行此操作，会检查指定的表空间是不是在同一节点组中定义的。

措施: 指定在包含要建立索引的文本列的表所在的节点组中定义的表空间。

CTE0232 指定的或缺省的表空间 “%1” 不是单节点的。这对于视图的索引或在指定 **CACHE TABLE** 选项时是必需的。

解释: 对于单个节点上的表，只有视图的索引或在启用了 **CACHE TABLE** 选项的情况下受支持。

措施: 如果缺省表空间导致此错误，则将表放置在单节点表空间中。或者，如果指定了多节点表空间，则指定另一单节点表空间。

CTE0233 正在运行的管理命令存在冲突。请稍后重试此命令。

解释: 另一管理命令仍在运行或异常终止而没有释放命令锁定。

措施: 使用 **CONTROL LIST** 检查哪些锁定仍是活动的。如果存在活动锁定而没有任何命令正在运行，则使用 **CONTROL CLEAR** 命令手工清除锁定。注意其他用户可能在运行持有锁定的管理命令。

CTE0234 正在对文本索引运行的管理命令存在冲突。请稍后重试此命令或指定 **DISABLE DATABASE** 命令的 **FORCE** 选项。

解释: 另一管理命令仍在运行或异常终止而没有释放命令锁定。

措施: 使用 **CONTROL LIST** 检查哪些锁定仍是活动的。如果存在活动锁定而没有任何命令正在运行，则使用 **CONTROL CLEAR** 命令手工清除锁定。注意其他用户可能在运行持有锁定的管理命令。对于 **DISABLE DATABASE** 命令，可以指定将停止对该数据库的所有其它命令的 **FORCE** 选项。

CTE0235 找不到 **DB2 Net Search Extender** 的任何有效许可证。

解释: 找不到 **DB2 Net Search Extender** 的任何有效许可证。

措施: 检查是否使用 **db2lic** 正确地安装了许可

证。确保在产品安装之后更新了现有实例。

CTE0236 对于 **MPP** 实例，只有 **Node0** 是受支持的。

解释: 如果具有要建立索引的文本列的表驻留在 **Node0** 上，则只能对 **MPP** 实例创建文本索引。

措施: 检查定义表的表空间的节点组。

CTE0237 内部错误: 日志表 “%1” . “%2” 包含无效操作 “%3”。

解释: 日志表记录对包含已索引文本列的表执行的操作。此表可能已损坏，原因是它包含并非由 **DB2 Net Search Extender** 写下的条目。

措施: 检查日志表，并删除损坏的条目。

CTE0238 内部错误: 表 “%1” . “%2” 在列 “%3” 中包含不正确的语法表达式。

解释: 表达式列表中的指定文本列存在错误。

措施: 检查定界符 “Begin” 和 “End” 对。

CTE0239 内部错误: 索引属性 “%1” 的总长度超出最大长度 “%2”。

解释: 超出索引属性的最大大小（1016 字节）。属性包含实例、索引、工作目录以及其它信息。

措施: 确保这些路径名不会过长。

CTE0240 内部错误: 设置环境变量 “%1” 失败。

解释: 设置指定的环境变量失败。环境设置可能存在问题。

措施: 检查 **OS** 特定基准。

CTE0241 内部错误: 数据链路
UDF “%1” . “%2” 返回类型
“%3”。期望的类型是: “%4”。

解释: 用来取装数据链路引用的内容的数据链路
UDF 是使用意外的数据类型定义的。

措施: 从 SYSCAT.FUNCTIONS 抽取数据链路 UDF
的定义, 并向 IBM 服务中心报告错误。

CTE0242 参数 “%2” 的值 “%1” 无效。

解释: 搜索存储过程或表值函数
DB2EXT.TEXTSEARCH 是使用无效参数调用的。

措施: 更正搜索存储过程或表值函数的参数值。有
关有效的参数, 参考文档。

CTE0243 文本索引 “%1” . “%2” 的高速缓
存尚未激活。

解释: Net Search Extender 操作需要激活的高速
缓存。高速缓存当前未激活。可能的原因包括:

- 在上一次 DB2TEXT START 命令之后从未激活高速缓存。
- 已使用 DB2TEXT DEACTIVATE CACHE 命令显式取消激活了高速缓存。

措施: 对索引执行 DB2TEXT ACTIVATE CACHE
命令, 并重新运行 Net Search Extender 操作。

CTE0244 内部错误: 对 “%1” 的调用返回
rc=“%2”, SQLCODE=“%3”。

解释: 调用内部函数时发生内部处理错误。

措施: 如果错误仍然存在, 则启动跟踪并检查
db2diag.log。报告错误。

CTE0245 请求的高速缓存大小超出可用的高速
缓存大小。将最大高速缓存大小增加
至大于 “%1” 的值或降低 pctfree
值。

解释: 装入所有数据所需的高速缓存大小超出索引
的 MAXIMUM CACHE SIZE 值。在激活高速缓存

(DB2TEXT ACTIVATE 命令) 期间或在激活高速
缓存时由索引更新操作检测到这一点。

措施: 如果在 DB2EXT ACTIVATE 命令中报告
了错误, 则使用 DB2EXT.MAXIMUM_CACHE_SIZE
函数重新计算最大高速缓存大小, 并改变索引的
MAXIMUM CACHE SIZE 设置。最后降低
PCTFREE 值。如果增量更新期间超出最大文档数,
则使用 db2 deactivate cache 和 db2text activate
cache recreate 命令重新构建高速缓存。

CTE0246 文件 “%1” 是空的。

解释: DB2TEXT CREATE INDEX 命令失败, 原
因是在命令中指定的文档模型文件是空的。

措施: 在命令中指定有效的文档模型文件。

CTE0247 未能创建 DB2 Net Search
Extender 存储过程。

解释: DB2TEXT ENABLE DATABASE 命令未能
创建内部存储过程 DB2EXT.CTESRVSP。

措施: 有关详细信息, 检查与 CREATE
PROCEDURE 语句相关联的附加 DB2 错误消息。
如果通过除去具有完全相同的名称的现有存储过程
不能更正错误, 则启动跟踪并报告错误。

CTE0248 生成的搜索字符串过长。降低搜索查
询的复杂性。

解释: Net Search Extender 查询太长或太复杂,
基本搜索引擎无法处理。复杂性受同义字比较、
FUZZY FORM OF 表达式和屏蔽字符的影响。

措施: 降低查询的复杂性或长度。

CTE0249 可执行程序 “%1” 异常终止。

解释: 执行 Net Search Extender 命令时, 调用了
可执行文件 “%1”, 但它异常终止。

措施: 验证可执行文件是否尚未通过用户交互作用
(例如, 信号) 显式终止。如果没有, 则启动跟踪,
重新运行命令, 并报告错误。

CTE0250 列类型变换函数 “%2” . “%3” 的
返回类型 “%1” 不受支持。

解释: 在 DB2TEXT CREATE INDEX 命令中, 指定了返回不受支持的数据类型的列类型变换。受支持的数据类型包括:
CHARACTER、VARCHAR、LONG
VARCHAR、CLOB、GRAPHIC、VARGRAPHIC、L
ONG VARGRAPHIC、DBCLOB、BLOB 和
ATALINK。

措施: 选择不同的列类型变换函数。

CTE0251 内部错误: 列类型 “%1” 不受支持。

解释: 使用的列类型不在受支持类型的列表中。

措施: 检查创建索引以获取 “键和建立索引” 的有效列的列表。对命令作适当更改并重试。如果仍然存在错误, 则启动跟踪并检查 db2diag.log。向 IBM 服务中心报告错误。

CTE0252 缺少参数 “%1”。

解释: 内部错误 — 执行 Net Search Extender 命令时, 在缺少参数 “%1” 的情况下调用管理可执行程序。

措施: 尝试更改 Net Search Extender 参数命令来避免问题。如果错误仍然存在, 则打开跟踪功能, 并向 IBM 服务中心报告错误。

CTE0253 找不到日志视图中列示的文档。

解释: 日志视图中列示的文本文档的内容已更改, 未能访问这些内容。

措施: 检查文档是否存在以及要包括在索引中的文本文档的读取 / 存取许可权。

CTE0254 索引 “%1” 的高速缓存已激活。

解释: 已使用 ACTIVATE CACHE 命令激活索引。

措施: 检查指定的索引名和正在使用的数据库。

CTE0255 缺少高速缓存结果列表 Expressions 的列名。
将 “AS <cache column name>”
添加至表达式。

解释: 必须命名高速缓存结果列表 Expressions。例如:
“C1+C2 AS myresult”。

措施: 将 “AS <cache column name>” 添加至表达式。

CTE0256 为建立索引选择数据所需的查询失败。降低属性、高速缓存表或初始搜索
结果次序 Expressions 的复杂性。

解释: Net Search Extender 从命令的表达式创建查询以便为建立索引而从数据库选择数据。查询失败, 原因是它太复杂。

措施: 降低属性、高速缓存表或初始搜索结果次序 Expressions 的复杂性。

CTE0257 创建共享内存时出错。

解释: 由于先前的错误或许可权问题, 未能创建共享内存资源。

措施: 检查 db2diag.log 以获取更多信息, 或者清除共享资源。另见错误 CTE0183。

CTE0258 共享内存版本错误。

解释: 未能存取共享内存资源, 原因是它已损坏或存在版本冲突。

措施: 有关进一步信息, 检查 db2diag.log。禁用并重新启用数据库, 然后重试。

CTE0259 不能在全局共享内存中插入条目。条
目已存在。

解释: 由于先前的错误, 要在全局共享内存中插入的条目已存在。

措施: 有关进一步信息, 检查 db2diag.log。使用 db2text stop 和 db2text start 命令重新启动更新和锁定服务。

CTE0260 不能存取全局共享内存中的条目。找不到条目。

解释: 由于先前的错误, 要从全局共享内存中除去的条目不存在。

措施: 有关进一步信息, 检查 db2diag.log。尝试使用 db2text stop 和 db2text start 命令重新启动更新和锁定服务。

CTE0261 对于此实例中的文本索引, 至少存在一个激活的高速缓存。使用 **DEACTIVATE CACHE** 命令取消激活任何已激活的索引的高速缓存, 或者使用 **FORCE** 选项来停止。

解释: 仅当对使用 **ACTIVATE CACHE** 命令激活的所有文本索引运行 **DEACTIVATE CACHE** 命令时, 才能使用 db2text stop 命令。

措施: 使用 **DEACTIVATE CACHE** 命令取消激活任何已激活的索引的高速缓存或使用 **FORCE** 选项来停止。

CTE0262 参数 “%1” 的值过长。

解释: 值超过最大允许大小。

措施: 检查最大大小。

CTE0263 文本索引 “%1” . “%2” 是使用 **RECREATE INDEX ON UPDATE** 选项创建的。在此上下文中, 可能未指定 **UPDATE MINIMUM** 或 **COMMITCOUNT FOR UPDATE**。

解释: 仅当以增量方式更新索引时, update minimum 和 commitcount for update 才有效。

措施: 如果想要每次执行更新时都重新创建索引, 则除去 **UPDATE MINIMUM** 和 **COMMITCOUNT FOR UPDATE** 设置。如果想要使用 **UPDATE MINIMUM** 和 **COMMITCOUNT FOR UPDATE**, 则不要指定 **RECREATE INDEX ON UPDATE**。

CTE0264 激活索引操作中发生了错误。有关详细信息, 检查事件视图 “%1” . “%2” 和 db2diag.log。

解释: 在索引激活过程期间, 错误会写至事件表和 db2diag.log 文件。

措施: 有关文档错误的更多信息, 检查事件表。在修正问题之后清除事件日志。

CTE0265 用户表或管理表空间 (“%1”) 的表空间不只是对节点 0 定义的。

解释: 如果文本索引是对 MPP 实例建立的, 则用户表的表空间一定只驻留在 Node0 上。

措施: 使用表空间驻留在 Node0 上的表。

CTE0266 **ValueFrom “%1” 一定要小于 ValueTo “%2”。**

解释: 属性搜索中指定的值无效。如果搜索语法为 “**BETWEEN ValueFrom AND ValueTo**”, 则下界 (ValueFrom) 必须小于上界 (ValueTo)。

措施: 更改 “**BETWEEN ValueFrom AND ValueTo**” 子句中的边界。

CTE0267 数据库 “%1” 中的 **Net Search Extender** 数据库对象处于不一致状态。

解释: 至少缺少或损坏了一个 DB2 Net Search Extender 对象。在安装新的 DB2 Net Search Extender 产品版本之后尚未迁移数据库, 或者数据库用户已更改或删除 Net Search Extender 内部对象。在这种情况下, 所有文本索引都会丢失, 数据库就不能再使用文本。

措施: 要将数据库迁移至当前版本, 请遵循 DB2 Net Search Extender 文档中的迁移描述。或者, 使用 **FORCE** 选项发出 **DB2TEXT DISABLE DATABASE** 命令。然后, 您可以通过使用 **DB2TEXT ENABLE DATABASE** 命令再次启用数据库的文本搜索。

CTE0270 在增量更新之后未能修改日志表“%1” . “%2”。下一次更新（UPDATE）期间要处理条目。

解释:

启动增量索引更新时，会创建时间戳记。它充当要处理的更改记录的阈值。于是与增量更新并行发生的更改将在以后进行处理（在下次更新期间）。在某些情况下，更新启动时未落实但在执行索引更新时落实的事务中可能有一些更改。这有可能导致不一致。

要避免这种不一致的情况，不要从日志表中删除阈值时间戳记之前的更改记录，即使它们只作了部分处理。在下次增量更新时，将对索引重新应用这些更改。

措施: 在下次索引更新时，会对索引重新应用这些更改。对于删除操作，这可能导致以下错误：CTE0101: IdEnReasonCode_Docmap_docid_not_found。

注意，可忽略此错误，原因是已删除文档。如果CTE0270 错误经常发生，则考虑删除索引并使用增量索引更新的已修改阈值重新创建索引。例如：
`db2text "CREATE INDEX ... INDEX CONFIGURATION(UPDATEDELAY 30)"`

这意味着增量更新期间处理只运行早于 30 秒的更改记录，而避免干预少于 30 秒的并行更改事务。

CTE0273 索引“%1”，“%2”的高速缓存已激活。

解释: 已使用 ACTIVATE CACHE 命令激活索引。

措施: 检查指定的索引名和正在使用的数据库。

CTE0274 连接的目标数据库系统“%1”不受支持。

解释: 已尝试在与 DB2 Net Search Extender 不支持的数据库系统连接的情况下执行 DB2TEXT 命令。

CTE0275 找不到服务器“%1”的类型和版本信息。

解释: 在 DB2 目录视图“SERVERS”中找不到服务器的类型和版本信息。

措施: 确保正确设置了 DB2 联合环境。

CTE0276 找不到数据链路 UDF。

解释: 未将数据链路 UDF 作为数据库中的用户定义函数注册。

措施: 在目录 db2ext/ddl 中执行命令“db2 -tvf ctedlcud.ddl”。

CTE0277 未能连接高速缓存存储段。

解释: 系统不能分配足够的内存来装入大型高速缓存段或打不开高速缓存段（原因是先前已删除它）。

措施: 检查系统设置并增加页面调度空间和可用内存量。对于大型高速缓存大小，您可能需要准备系统。参考 DB2 Net Search Extender 文档。使用 DEACTIVATE 和 ACTIVATE [RECREATE] 命令来重新创建高速缓存。如果问题仍然存在，则检查 db2diag.log 以获取附加信息。

CTE0278 在 AIX 32 位系统上，在激活大型高速缓存之前更改 MAXDATA 设置。

解释: 当在 AIX 32 位系统上使用搜索存储过程时，可能需要更改 db2fmp 可执行文件的 MAXDATA 设置。

措施: 有关更改 MAXDATA 设置的详细信息，参考 DB2 Net Search Extender 文档。

CTE0279 高速缓存数据的大小已达到系统限制。

解释: 通过降低 PCTFREE 值，可以增加高速缓存激活期间的最大数据大小。这使系统能够在高速缓存中保留较少的可用空间。

措施: 使用较低的 PCTFREE 值或减少要高速缓存的文本数据量。使用 DEACTIVATE 和 ACTIVATE [RECREATE] 命令来重新创建高速缓存。

CTE0280 没有足够的磁盘空间来写入持久高速缓存文件。

解释: 系统不能在高速缓存目录中为持久高速缓存写入足够大的文件。

措施: 使用 ALTER INDEX 命令, 将持久高速缓存目录更改为空的文件系统。或者通过降低 PCTFREE 或 MAXIMUM CACHE SIZE 值或通过使用临时高速缓存来减少高速缓存大小。

CTE0281 删除持久高速缓存文件 “%1” 失败。

解释: 文件不存在或不能存取。

措施: 检查此文件是否仍然存在并手工删除它。

CTE0282 高速缓存中的文档数已达到系统限制。

解释: 通过降低 PCTFREE 值, 可以增加高速缓存激活期间要高速缓存的最大文档条目数。这使系统能够在高速缓存中保留较少的可用空间。

措施: 使用较低的 PCTFREE 值或减少高速缓存中的文档条目的量。使用 DEACTIVATE 和 ACTIVATE [RECREATE] 命令来重新创建高速缓存。

CTE0283 未能创建高速缓存存储段。

解释: 系统不能为将大型高速缓存段装入到内存中分配足够内存。通过降低 PCTFREE 值, 可达到较小的高速缓存段大小。

措施: 检查系统设置并增加页面调度空间和可用内存量。可以通过使用较低的 PCTREE 值来降低高速缓存大小。对于大型高速缓存大小, 可能需要准备系统。参考 DB2 Net Search Extender 文档。使用 DEACTIVATE 和 ACTIVATE [RECREATE] 命令来重新创建高速缓存。如果问题仍然存在, 则检查

db2diag.log 以获取附加信息。

CTE0284 文本索引位于节点 “%1” 上, 但是对节点 “%2” 调用了搜索函数。

解释: 未对索引所在的节点调用搜索存储过程或表值函数 DB2EXT.TEXTSEARCH。搜索函数将不会自动分布至正确节点。

措施: 将 DB2NODE 环境变量设置为在连接数据库之前连接索引的节点。

CTE0285 对于分布至多个节点的文本索引, 不允许搜索函数。

解释: 一定不要使用分布至多个节点的索引调用表值函数 DB2EXT.TEXTSEARCH, 原因是它将不会自动分布至正确节点, 而是在协调程序节点上执行。

措施: 在多节点环境中使用 CONTAINS、SCORE 或 NUMBEROFMATCHES 函数。

CTE0286 在 “%1” . “IBMSNAP_REGISTER” 中找不到源表 “%2” . “%3” 和捕获更改表 “%4” . “%5” 的任何行。

解释: 对于在 DB2TEXT CREATE INDEX 命令中指定的复制捕获表特征, 在 IBMSNAP_REGISTER 表中找不到任何有效条目。对于索引列 SOURCE_OWNER 和 SOURCE_NAME (SOURCE_VIEW_QUAL=0) 以及列 PHYS_CHANGE_OWNER 和 PHYS_CHANGE_TABLE 中的指定复制捕获表, 有效条目必须包含指定源表。

可能原因: 指定源表未注册为复制捕获表的复制源。

措施: 为 DB2 复制正确注册源表, 或为源表指定正确的复制捕获表。

CTE0287 对于源表 “%4” . “%5” 和捕获更改表
 “%6” . “%7” , “%3” . “IBM
 SNAP_REGISTER” 中的 “%2” 的
 值 “%1” 无效。

解释: 不允许 IBMSNAP_REGISTER 表中的复制设置。可能原因: 1. 列 CHG_UPD_TO_DEL_INS 未包含值 “Y”。2. 列 CCD_CONDENSED 包含值 “Y”。

措施: 当为 DB2 复制注册源表时, 确保将更新操作变换为删除和插入操作对。此外, 还应确保没有使用任何压缩复制捕获表。

CTE0288 源表 “%1” . “%2” 和捕获更改表
 “%3” . “%4” 在不同的服务器
 (“%5” 和 “%6”) 上。

解释: 指定的源表和复制捕获表必须驻留在同一服务器上。

CTE0289 包装器 “%1” 不受支持。

解释: 包装器不受支持。有关受支持包装器的列表, 参考 DB2 Net Search Extender 文档。

CTE0290 在复制子句中不允许别名
 “%1” . “%2”。

解释: 不允许在复制子句中对昵称指定别名。

措施: 指定昵称而不是别名或为远程表创建新昵称。

CTE0451 突出显示 UDF 不支持指定的文档格式 “%1”。

解释: 文档格式 “%1” 不支持突出显示。

措施: 使用突出显示 UDF 支持的文档格式。

CTE0452 突出显示 UDF 中的选项 “%1” 附近存在语法错误。

解释: 在指定选项附近指定了不正确的语法。

措施: 检查选项 %1 附近的语法。更正错误并重试。

CTE0453 突出显示 UDF 的返回大小过小。

解释: 突出显示文档的请求部分无法装入到突出显示 UDF 的返回参数中。

措施: 降低窗口编号、窗口大小和 / 或应该显示点击的节数。这将减少返回给用户的文档部分。

CTE0454 将突出显示 UDF 的参数从代码页
 “%1” 转换为代码页 UTF8 时发生了错误。

解释: 指定 CCSID (它可能是缺省数据库 CCSID) 中的突出显示 UDF 的参数不能转换为 UTF8。

措施: 确保正确地指定 CCSID。

CTE0455 数据库代码页 “%1” 在突出显示
 UDF 中不受支持。

解释: 突出显示 UDF 不支持数据库的代码页。

CTE0456 突出显示 UDF 只支持代码页 UTF8
 中的文档。

解释: 只有代码页 UTF8 中的文档才支持突出显示 UDF。

CTE0457 参数 “%2” 的值 “%1” 在突出显示
 UDF 中无效。

解释: 突出显示参数的值无效。

措施: 检查参数值, 并确保在数据范围内允许该值。

CTE0841 缺少命令选项 “%1”。

解释: 未指定必需的命令选项。

措施: 检查指定的参数并添加缺少的参数。

CTE0842 没有为命令选项 “%1” 指定任何值。

解释: 未指定命令选项的必需值。

措施: 检查指定的参数并添加缺少的选项。

CTE0843 没有为命令选项 “%1” 指定任何数字值。

解释: 指定了字符串而不是数字。

措施: 检查指定的参数并将字符串更改为正确的数字。

CTE0844 定义文件路径 “%1” 过长。

解释: 指定的路径过长因而未能处理。

措施: 使用较短的路径并重试。

CTE0845 没有指定任何定义文件。

解释: 需要指定定义文件。

措施: 添加有效定义文件并再次尝试调用。

CTE0846 定义文件名 “%1” 过长。

解释: 指定的定义文件名过长。

措施: 将定义文件名的长度减少为允许的大小。

CTE0847 定义文件 “%1” 不存在。

解释: 找不到指定的定义文件。

措施: 检查定义文件是否在正确的路径中以及是否可供当前用户存取。

CTE0849 未能锁定字典文件 “%1”。

解释: 进程无法锁定字典文件。您没有写存取权或另一进程已打开文件以进行写操作。

措施: 检查运行的进程以确保没有任何进程正锁定字典文件，并检查您的存取权限。

CTE0850 输出文件 “%1” 已存在。

解释: 未能覆盖指定的输出文件。

措施: 检查您能否在指定目录中创建同义字。

CTE0851 字典文件 “%1” 不完整。

解释: 同义字典文件已损坏。

措施: 清除目录并再次编译定义文件。

CTE0852 字典文件 “%1” 版本错误。

解释: 字典文件是使用同义字编译器的较早版本生成的。

措施: 使用同义字编译器的最新版本再次编译定义文件。

CTE0853 不能覆盖现有字典 “%1”。

解释: 不能覆盖现有字典。

措施: 检查对字典文件的写存取权、它的目录位置和子目录位置。

CTE0855 未正确指定同义字术语。

解释: 定义文件中存在语法错误。

措施: 有关创建同义字定义文件和同义字支持的信息，检查 DB2 Net Search Extender 文档。

CTE0856 定义文件 “%1” 是空的。

解释: 不允许空的定义文件。

措施: 有关创建同义字定义文件和同义字支持的信息，检查 DB2 Net Search Extender 文档。

CTE0857 在文件 “%1” 的第 “%2” 行中找不到任何块起始行。

解释: 定义文件中存在语法错误。

措施: 块必须以 “:WORDS” 开始。有关同义字概念的信息，检查 DB2 Net Search Extender 文档。

CTE0858 在文件 “%1” 的第 “%2” 行中指定了无效关系。

解释: 定义文件中存在语法错误。

措施: 必须检查 “相关联的术语定义”。有关创建同义字定义文件的信息，检查 DB2 Net Search Extender 文档。

CTE0859 关系编号不在文件 “%1” 的第 “%2” 行的范围内。

解释: 用户定义关系都是以相关联的类型为基础的。它们由 1 与 128 之间的唯一数字标识。

措施: 验证关系编号。

CTE0861 没有在文件 “%1” 的第 “%2” 行中定义任何术语。

解释: 未指定必需术语。

措施: 有关创建同义字定义文件的信息，检查 DB2 Net Search Extender 文档。

CTE0861 文件 “%1” 的第 “%2” 行中的同义字术语过长。

解释: 同义字术语的长度限制为 64 个字节。

措施: 改变同义字术语的大小并重试。

CTE0862 在文件 “%1” 的第 “%2” 行中不正确地指定了强度。

解释: 定义文件中存在语法错误。

措施: 有关创建同义字定义文件和同义字支持的信息，检查 DB2 Net Search Extender 文档。

CTE0863 强度不在文件 “%1” 的第 “%2” 行的范围内。

解释: 强度值应指定为 1 与 100 之间。

措施: 更改强度值以使它为从 1 至 100 的数字值。

CTE0864 内部错误: 同义字编译器失败，原因代码为 “%1”。

解释: 发生了内部处理错误，不允许进一步处理。尝试启动并停止更新和锁定服务以及 DB2。

措施: 如果错误仍然存在，则启动跟踪并检查 db2diag.log。

CTE0865 未能创建目录 “%1”。

解释: 未能创建指定的目录。

措施: 检查目录是否已经存在以及针对该目录的许可权。

CTE0866 未能除去目录 “%1”。

解释: 未能除去目录。

措施: 检查您是否对指定的目录具有写许可权。

附录 G. 文档模型参考

DB2 Net Search Extender 提供文档模型的下列参考信息:

- 文档模型的 DTD
- 定位器 (XPath) 表达式的语义
- 文本字段和文档属性的限制
- Outside-In 标记属性值

文档模型的 DTD

以下是文档类型定义 (DTD) 形式的文档模型语法的正式描述:

```
<!ELEMENT GPPModel (GPPFieldDefinition|GPPAttributeDefinition)+>
<!ELEMENT HTMLModel (HTMLFieldDefinition|HTMLAttributeDefinition)+>
<!ELEMENT XMLModel (XMLFieldDefinition|XMLAttributeDefinition)+>

<!ELEMENT GPPFieldDefinition EMPTY>
<!ATTLIST GPPFieldDefinition name CDATA #REQUIRED>
<!ATTLIST GPPFieldDefinition start CDATA #REQUIRED>
<!ATTLIST GPPFieldDefinition end CDATA #IMPLIED>
<!ATTLIST GPPFieldDefinition exclude (YES|NO) NO>

<!ELEMENT GPPAttributeDefinition EMPTY>
<!ATTLIST GPPAttributeDefinition name CDATA #REQUIRED>
<!ATTLIST GPPAttributeDefinition start CDATA #REQUIRED>
<!ATTLIST GPPAttributeDefinition end CDATA #REQUIRED>
<!ATTLIST GPPAttributeDefinition type NUMBER #REQUIRED>

<!ELEMENT HTMLFieldDefinition EMPTY>
<!ATTLIST HTMLFieldDefinition name CDATA #REQUIRED>
<!ATTLIST HTMLFieldDefinition tag CDATA #REQUIRED>
<!ATTLIST HTMLFieldDefinition meta-qualifier CDATA #IMPLIED>
<!ATTLIST HTMLFieldDefinition exclude (YES|NO) NO>

<!ELEMENT HTMLAttributeDefinition EMPTY>
<!ATTLIST HTMLAttributeDefinition name CDATA #REQUIRED>
<!ATTLIST HTMLAttributeDefinition tag CDATA #REQUIRED>
<!ATTLIST HTMLAttributeDefinition meta-qualifier CDATA #IMPLIED>
<!ATTLIST HTMLAttributeDefinition type NUMBER #REQUIRED>

<!ELEMENT XMLFieldDefinition EMPTY>
<!ATTLIST XMLFieldDefinition name CDATA #REQUIRED>
<!ATTLIST XMLFieldDefinition locator CDATA #REQUIRED>
<!ATTLIST XMLFieldDefinition ignore (YES|NO) NO>
<!ATTLIST XMLFieldDefinition priority CDATA #IMPLIED>
<!ATTLIST XMLFieldDefinition exclude (YES|NO) NO>
```

```
<!ELEMENT XMLAttributeDefinition EMPTY>
<!--ATTLIST XMLAttributeDefinition name CDATA #REQUIRED-->
<!--ATTLIST XMLAttributeDefinition locator CDATA #REQUIRED-->
<!--ATTLIST XMLAttributeDefinition ignore (YES|NO) NO-->
<!--ATTLIST XMLAttributeDefinition priority CDATA #IMPLIED-->
<!--ATTLIST XMLAttributeDefinition type NUMBER #REQUIRED-->
```

定位器（XPath）表达式的语义

根据 XML 数据模型，XML 文档被视作包含下列类型节点的树：

- 根节点
- 元素节点
- 文本节点
- 属性节点
- 名称空间节点
- 处理指令节点
- 注释节点

这些节点之间的链接（换句话说就是树形成关系）反映 XML 文档中的直接包含关系。

根节点只能出现在根处，而不能出现在树的其它位置。它包含作为其子代的文档元素和可选注释及处理指令。

元素节点可包含任何类型的节点，但根节点除外。其它类型的节点只允许在树的终端节点位置。

有三种类型的**包含链接**：“子代”、“属性”和“名称空间”。“属性”和“名称空间”包含链接必须分别引导至属性和名称空间节点。换言之，要存取元素节点的子代（按照图论），需要跟随“属性”链接来查找所有包含的属性，跟随“名称空间”链接来查找所有包含的名称空间声明，跟随“子代”链接来查找包含的元素、文本节点、处理指令和注释。

需要根据上下文节点来解释 XPath 表达式，并表示一组节点。当用作 Net Search Extender 选择器模式时，上下文节点空闲，也就是说，相对路径模式 *p* 解释为 *//p*。

下面是 Net Search Extender XPath 选择器模式：

- 上下文 *N* 中的 Pattern '*|*' LocationPathPattern 表示 Pattern 和 LocationPathPattern（都在上下文 *N* 中）与之相匹配的节点的并集。

- 上下文 N 中的 `'/'RelativePathPattern` 表示此 `RelativePathPattern` 在根上下文中表示的任何内容。
- 上下文 N 中的 `'/'/'RelativePathPattern` 表示在作为根的子代（在子代轴上）的任何上下文中解释的这一 `RelativePathPattern` 表示法的并集。
- 当且仅当上下文 N 中的节点与其父代的上下文中的 `StepPattern` 相匹配，且其父节点与上下文 N 中的 `RelativePathPattern` 相匹配时，`RelativePathPattern '/' StepPattern` 才与该节点相匹配。
- 当且仅当上下文 N 中的节点与其父代的上下文中的 `StepPattern` 相匹配，且它的祖先节点与上下文 N 中的 `RelativePathPattern` 相匹配时，`RelativePathPattern '/'/' StepPattern` 才与该节点相匹配。
- 上下文 N 中的 `'child'::NodeTest`（缩写语法：`NodeTest`）与作为 N 的子代（在子代轴上）且符合 `NodeTest` 的节点相匹配。
- 上下文 N 中的 `'attribute'::NodeTest`（缩写语法：`@NodeTest`）与作为 N 的属性且符合 `NodeTest` 的节点相匹配。
- 当且仅当节点为指定类型时，`NodeType '(' ')' '` 才适合该节点。
- `'processing-instruction' '(' Literal ')' '` 适合将 `Literal` 作为其名称的任何处理指令类型节点。
- `'*'` 适合任何元素或属性节点（元素名称的名称屏蔽）。
- `NCName ':' '*'` 适合将 `NCName` 作为其名称前缀的任何元素节点。
- `QName` 适合具有指定名称的任何节点。

注意

格式为 `NameTest` 的 `NodeTest` 假定节点为所选轴上的主体类型，它是属性轴上的属性类型，也是子代轴上的子类型。因此，`NameTest` 不能用来选择注释或处理指令节点，而只能选择子节点和属性节点。而且，该模式允许选择任何类型的节点（名称空间节点除外），原因是不允许使用轴说明符“`namespace`”。

模式示例:

- `chapter | appendix` 表示所有章节元素和附录元素
- `table` 表示所有表元素
- `*` 表示所有元素（注意，这是子元素 `::*` 的缩写）
- `ulist/item` 表示具有 `ulist` 父代的所有术语元素
- `appendix//subsection` 表示具有附录祖先的所有子节元素

- / 表示只包含根节点的单元素集合
- comment() 表示所有注释节点
- processing-instruction() 表示所有处理指令
- attribute::* (或 @*) 表示所有属性节点

以下是定位器元素的语法:

```
Locator      ::= LocationPathPattern
              | Locator '|' LocationPathPattern
LocationPathPattern ::= '/' RelativePathPattern ?
              | '///'? RelativePathPattern
RelativePathPattern ::= StepPattern
                    | RelativePathPattern '/' StepPattern
                    | RelativePathPattern '///' StepPattern
StepPattern      ::= ChildOrAttributeAxisSpecifier NodeTest
ChildOrAttributeAxisSpecifier ::=
    ('child' | 'attribute') '::'
    | '@'?
NodeTest         ::= NameTest
                    | NodeType '(' ')'
                    | 'processing-instruction' '(' Literal ')'
NameTest         ::= '*' | NCName ':' '*' | QName
NodeType         ::= 'comment' | 'processing-instruction'
```

NCName 和 QName 是按“XML 名称建议”定义的:

文本字段和文档属性的限制

以下是文本字段和文档属性的限制的列表:

- 索引中的字段的最大数目: 32767
- 一个文档中类型为 STRING 的一个属性的值的最大数目: 1024
- 类型为 STRING 的属性的最大数目: 253
- STRING 属性值中的字符数截断为 128
- 类型为 DATE 和 NUMBER 的属性的最大数目: 32766
- DATE 或 NUMBER 属性值中的字符数截断为 128
- 对于 NUMBER 属性, 接受双精度浮点数作为值。
- 可对一个文档中类型为 DATE 或 NUMBER 的一个属性指定的值的最大数目: 无限制

下列各项是可包括在 HTML 文档模型中的标记:

- <A>
- <ADDRESS>
- <AU>

- <AUTHOR>
- <H1>
- <H2>, <H3>, <H4>, <H5>
- <H6>
- <TITLE>

不能在 HTML 文档模型中将可包含其它标记的标记（如 <HEAD> 和 <BODY>）指定为文本字段。

Outside-In 标记属性值

与 Outside-In 文档属性标记类型相关的标记属性的可能值:

SCCC_A_ABSTRACT
 SCCC_A_ACCOUNT
 SCCC_A_ADDRESS
 SCCC_A_ATTACHMENTS
 SCCC_A_AUTHORIZATION
 SCCC_A_BACKUPDATE
 SCCC_A_BASEFILELOCATION
 SCCC_A_BILLTO
 SCCC_A_BLINDCOPY
 SCCC_A_CARBONCOPY
 SCCC_A_CATEGORY
 SCCC_A_CHECKEDBY
 SCCC_A_CLIENT
 SCCC_A_COMPANY
 SCCC_A_COMPLETEDDATE
 SCCC_A_COUNTCHARS
 SCCC_A_COUNTPAGES
 SCCC_A_COUNTWORDS
 SCCC_A_CREATIONDATE
 SCCC_A_DEPARTMENT
 SCCC_A_DESTINATION
 SCCC_A_DISPOSITION
 SCCC_A_DIVISION
 SCCC_A_DOCCOMMENT
 SCCC_A_DOCTYPE
 SCCC_A_EDITMINUTES
 SCCC_A_EDITOR
 SCCC_A_FORWARDTO
 SCCC_A_GROUP
 SCCC_A_KEYWORD
 SCCC_A_LANGUAGE
 SCCC_A_LASTPRINTDATE
 SCCC_A_LASTSAVEDBY
 SCCC_A_MAILSTOP
 SCCC_A_MANAGERSCCC_A_MATTER
 SCCC_A_OFFICE
 SCCC_A_OPERATOR
 SCCC_A_OWNER

SCCCA_PRIMARYAUTHOR
SCCCA_PROJECT
SCCCA_PUBLISHER
SCCCA_PURPOSE
SCCCA_RECEIVEDFROM
SCCCA_RECORDEDBY
SCCCA_RECORDEDDATE
SCCCA_REFERENCE
SCCCA_REVISIONDATE
SCCCA_REVISIONNOTES
SCCCA_REVISIONNUMBER
SCCCA_SECONDARYAUTHOR
SCCCA_SECTION
SCCCA_SECURITY
SCCCA_SOURCE
SCCCA_STATUS
SCCCA_SUBJECT
SCCCA_TITLE
SCCCA_TYPIST
SCCCA_USERDEFINEDPROP
SCCCA_VERSIONDATE
SCCCA_VERSIONNOTES
SCCCA_VERSIONNUMBER

与 Outside-In 开始和结束标记子类型相关的标记属性的可能值:

SCCCA_ALTFONTDATA
SCCCA_ANNOTATIONREFERENCE
SCCCA_CAPTIONTEXT
SCCCA_CHARACTER
SCCCA_COMPILEDFIELD
SCCCA_COUNTERFORMAT
SCCCA_CUSTOMDATAFORMAT
SCCCA_DATEDEFINITION
SCCCA_DOCUMENTPROPERTYNAME
SCCCA_ENDNOTEREFERENCE
SCCCA_FONTANDGLYPHDATA
SCCCA_FOOTNOTEREFERENCE
SCCCA_FRAME
SCCCA_GENERATEDFIELD
SCCCA_GENERATOR
SCCCA_HYPERLINK
SCCCA_INDEX
SCCCA_INDEXENTRY
SCCCA_INLINEDATAFORMAT
SCCCA_LISTENTRY
SCCCA_MERGEENTRY
SCCCA_NAMEDCELLRANGE
SCCCA_REFERENCEDTEXT
SCCCA_STYLE
SCCCA_SUBDOCTEXT
SCCCA_TOA
SCCCA_TOAENTRY
SCCCA_TOC

SCCCA_TOENTRY
SCCCA_TOF
SCCCA_VECTORSVETAG
SCCCA_XREF

注意，这些表包括任何文档属性以及 INSO 过滤器识别的所有标记子类型。有两个子类型例外：SCCCA_DOCUMENTPROPERTY 和 SCCCA_BOOKMARK。

附录 H. 文本搜索引擎

DB2 Net Search Extender 提供以下“文本搜索引擎”信息：

- 标记化
- 无用词

标记化

在建立索引期间，Net Search Extender 用以下方法处理文档文本：将文本分成几个标记。

词语

使用所有字母数字字符（“a”至“z”、“A”至“Z”和“0”至“9”）来创建全文索引。隔离字符是空白字符和下面的语句识别部分中描述的字符。控制字符（例如，换行（又称为换行符）和空白字符）解释如下：行中的控制字符（小于 0x20）被视作空白字符。忽略换行（0x0A）之前和之后的空白字符和控制字符。将 1 字节字符前后的换行视作是空白字符，而始终将表示同一字符的 2 字节字符视作相同字符。如果在搜索期间未指定任何内容，则将同一字符的大写字母和小写字母（例如，“A”和“a”）视作相同字符；如果在搜索期间需要精确匹配，则将它们视作不同字符。

句子

Net Search Extender 将后跟空白字符的“!”、“?”和“.”以及行末尾处的日语和中文句点识别为句子结束。

段落

段落识别与文档格式有关。在纯文本格式中，任何两个连续的换行符（可能带有插入回车）被识别为段落边界。在 HTML 中，段落标记 <p> 解释为段落边界。其它文档格式不支持段落识别。

无用词

无用词是出现频率很高的词语，且没有文本检索过程的相关内容。通常所有函数词语（在语言敏感环境中）都被认为是无用词，例如，“and”、“or”和“in”。

Net Search Extender 为语言列表提供无用词处理，其中无用词是未建立索引的，因此不能搜索它们。但是，无用词处理的结果是更小更快的文本索引。

注意，处理未建立索引的无用词的方式与搜索期间处理正常词语的方式相同。但是，如果对无用词建立了索引，则在搜索过程期间会忽略该无用词。

语言支持无用词

下列语言提供无用词处理。

AR_AA	在阿拉伯国家或地区使用的阿拉伯语
CA_ES	在西班牙使用的加泰罗尼亚语
DA_DK	在丹麦使用的丹麦语
DE_CH	在瑞士使用的德语
DE_DE	在德国使用的德语
EL_GR	在希腊使用的希腊语
EN_GB	在英国使用的英语
EN_US	在美国使用的英语
ES_ES	在西班牙使用的西班牙语
FI_FI	在芬兰使用的芬兰语
FR_CA	在加拿大使用的法语
FR_FR	在法国使用的法语
HE_IL	在以色列使用的希伯来语
IS_IS	在冰岛使用的冰岛语
IT_IT	在意大利使用的意大利语
IW_IL	在以色列使用的希伯来语
NB_NO	在挪威使用的挪威博克马尔语
NL_BE	在比利时使用的荷兰语
NN_NO	在挪威使用的挪威尼诺斯克语
PT_BR	在巴西使用的葡萄牙语
PT_PT	在葡萄牙使用的葡萄牙语
RU_RU	在俄罗斯使用的俄语
SV_SE	在瑞典使用的瑞典语

附录 I. 文本搜索引擎原因代码

0	操作成功执行 — 没有发生任何错误。
1	将无效句柄传递至函数。
2	函数未能分配足够的内存。
3	由于存取限制或安全性限制，所以函数未能执行。
4	此版本的“文本搜索引擎”运行时不支持该操作。
5	当前未启用该操作。
6	应用程序以非法次序调用“文本搜索引擎”函数，违反了“文本搜索引擎”协议。
7	发生了意外错误。请向服务代表报告此错误。
8	指定了无效语言。
9	指定的语言有效，但“文本搜索引擎”运行时不支持它。
10	指定了无效 CCSID。
11	指定的 CCSID 有效，但“文本搜索引擎”运行时不支持它。
12	指定了无效文档标识。
13	指定的文档格式有效，但“文本搜索引擎”运行时不支持它。
14	指定了无效文档格式。
15	由于文件输入 / 输出期间的存取限制，所以该操作失败。
16	由于文件输入 / 输出期间发生读取错误，所以该操作失败。
17	由于文件输入期间发生读取错误，所以该操作失败。
18	由于文件输出期间发生写错误，所以该操作失败。
19	由于文件输入 / 输出期间发生寻道错误，所以该操作失败。
20	由于文件输入 / 输出期间发生通知错误，所以该操作失败。
21	由于文件输入 / 输出期间发生关闭错误，所以该操作失败。
22	由于重命名操作期间发生错误，所以该操作失败。
23	由于除去操作期间发生错误，所以该操作失败。
24	由于 mkdir 操作期间发生错误，所以该操作失败。

- 25** 一个或多个函数自变量确实具有无效值（例如，意外的空指针或无效枚举类型值）。
- 26** 指定的目录不存在。
- 27** 发生了意外的“文本搜索引擎”错误。有关进一步的详细信息，请查看错误信息对象中的“文本搜索引擎”错误代码。
- 28** 发生了意外的 COS 错误。请报告此错误。
- 29** 已尝试更新空文档。
- 30** 此操作不支持指定的自变量。
- 31** 尝试分析日期属性时，日期属性解析器发现无效值。
- 32** 尝试分析编号属性时，编号属性解析器发现无效值。
- 33** 属性名无效，可能是名称过长。
- 35** 保留编号以备将来使用。
- 36** 输入文档包含的属性（DATE、NUMBER 或 STRING）超出属性的长度限制。该属性文本已根据此限制被截断。
- 38** 已超出用户设置的警告阈值。因此生成此错误。
- 39** 未能对输入文档建立索引。它包含过多的嵌套字段。
- 40** 已超出此索引的某个属性类型的不同属性的限制。
- 46** 迭代符不（不再）是有效的，原因是它的列表是空的或已删除。
- 47** 传递的句柄类型不支持该函数。例如，当尝试对并不表示查询结果迭代符的列表迭代符使用 `itlQueryResultEntryObtainData` 时，会发生此错误。
- 48** 如果针对指定语言和资源路径找不到无用词文件，则会发出此警告。
- 49** 如果无用词文件未包含任何无用词，则会发出此警告。
- 50** 如果无用词文件确实包含无效数据，则会发出此警告。
- 100** 因为具有指定名称和 / 或目录的索引不存在，所以未能打开它。
- 101** 指定的索引名是无效的索引名。
- 102** 指定的索引目录是无效的目录名。
- 103** 因为“文本搜索引擎”检测到索引结构和 / 或索引文件集已损坏，所以不能执行该操作。
- 104** 因为具有指定名称和目录的指定索引已存在，所以不能创建它。

109	必须执行回滚操作才能对此索引执行任何其它操作。
110	索引配置文件未包含在错误上下文中指定的必需部分。
111	索引配置文件未包含在错误上下文中指定的必需选项。
112	索引配置文件在错误上下文中指定的选项中包含无效数据。
113	索引配置文件与“文本搜索引擎”版本不匹配。
200	指定的文档模型名是无效的模型名。
201	指定的文档模型字段名是无效的字段名。
202	指定的文档模型未知。
203	指定的文档模型已存在，并且不能重新定义它。
204	已向索引添加了过多或过大的文档模型。
205	文档模型包含过多元素。
206	文档模型元素包含此类型元素不允许的参数（XML 属性）。
207	文档模型元素包含此类型参数（XML 属性）不允许的参数值。
208	文档模型元素未包含必需的参数（XML 属性），例如，“name”。
209	文档模型看起来不是 XML，或者它以意外的 XML 元素开头。
210	给定的 XPath（定位器值）包含意外的标记。
211	给定的 XPath（定位器值）包含意外的轴说明符（名称后跟两个冒号）。
212	给定的 XPath（定位器值）包含意外的节点测试。
213	文档模型目录文件（extension .mdx）已损坏。
214	文档模型索引文件（extension .mox）已损坏。
215	文档包含映射至文档属性的 XML 元素，并且该元素包含另一个文档属性。忽略内部属性。
216	给定的参数值作为 GPP 或 HTML 标记过长。
217	文档模型包含重复的字段定义。
218	文档模型包含重复的属性定义。
300	因为“文本搜索引擎”检测到用于文档名称映射的索引文件中有损坏，所以不能执行该操作。
301	因为“文本搜索引擎”检测到无效文档号，所以不能执行该操作。

302	因为“文本搜索引擎”检测到无效文档标识符，所以不能执行该操作。
303	因为“文本搜索引擎”找不到文档标识符的任何索引条目，所以不能执行该操作。
304	因为“文本搜索引擎”找不到文档号的任何索引条目，所以不能执行该操作。
305	因为“文本搜索引擎”检测到使用的文档号中存在溢出，所以不能执行该操作。
306	应用程序尝试对其建立索引的文档标识符已出现在文档列表中。 “文本搜索引擎”不支持重复的文档标识符出现在一个索引序列中（即，在更新落实之前）。
340	术语强度无效。
341	相关编号无效，必须在有效范围内。
342	相关类型无效，应使用 API 中描述的一个定义。
343	短语（术语）过长。
344	读取时遇到文件的意外结束。
345	读取索引 / 同义字文件时检测到版本冲突。
346	同义字缓冲区中存在溢出。
347	文件或目录名无效，可能是名称过长。
348	搜索没有发现字典中的术语（短语）或定义文件中的条目不包含必需的术语。
349	定义文件是空的。
350	通过输入参数指定的同义字典或定义文件不存在。
351	定义文件中存在语法错误。
352	未正确指定关系。
352	关系编号超出范围。
360	使用了无效的单字符屏蔽。
361	使用了无效的多字符屏蔽。
362	运算符数量小于查询中给定的操作数的数目。
363	运算符值超出 ItlEnOperator 枚举定义的范围。
364	排列公式的值超出枚举范围。

365	标识近似匹配段的编号超出范围。
366	查询正在构造中，不能重定义或复位。
367	作为先前搜索结果给定的作用域指示空结果。
368	在设置第一个字段名之前请求添加字段名的调用无效。
369	忽略请求与索引内容的无效比较的无效搜索标志。例如，如果对以不区分大小写的方式构建的索引请求了区分大小写比较，则会在错误信息中显示此原因代码。
370	泰国语或 DBCS 语言不支持字符串的屏蔽。
371	没有任何有效查询输入。例如，搜索术语可用。
372	请求了无效比较操作。
373	请求了无效比较操作。
374	对空索引请求了搜索索引句柄。
375	运算符和请求的运算符方式的组合不受支持。
380	搜索结果不完整，搜索因为阈值关系而没有继续。
381	索引查询发现查询包含无用词。
401	因为“文本搜索引擎”检测到用于字段 / 属性名称映射的索引文件中有损坏，所以不能执行该操作。
402	因为“文本搜索引擎”检测到无效字段或属性名，所以不能执行该操作。
403	因为给定的字段或属性名未知，所以不能执行该操作。
404	已超出此索引的某个属性类型的不同属性或不同字段的限制。
500	文档 / 数据包含无效字符序列（在 UTF8、UTF16 或 DBCS 源代码中）。
501	代码页转换器出错。
502	文档 / 数据包含不完整的字符序列（在 UTF8、UTF16 或 DBCS 源代码中）。
503	代码页转换器具有无效描述符。
600	XML 文档包含异步实体。例如，未用引号引起来的 XML 属性值。
602	无效字符引用（例如，or）。
603	无效二进制实体引用。

- 604** 未能创建 XML 解析器 Expat。
- 605** 标记中的属性名必须唯一。
- 607** XML 解析器发现无效外部实体引用。
- 608** 文档包括不正确的标记（例如，缺少 < 或 >）。
- 609** XML 文档必须具有封闭标记，在此封闭结束标记之后不允许出现任何文本。
- 610** 不允许处理指令出现在其位置上。例如，第一个处理指令不是 prolog <?xml .. ?>。
- 611** 元素的先后顺序是开始标记、内容和结束标记。例如，序列 “<s> text /s>” 中发生此错误，原因是结束标记不正确。
- 612** XML 解析器中的内存分配失败。
- 614** 无效参数实体引用。
- 615** 不完整字符，可能只是 2 字节 UTF8 字符的第一个字节。
- 616** 递归实体引用。
- 617** XML 语法错误；例如，文本超出封闭开始标记和结束标记。
- 618** 每一个开始标记都需要相匹配的结束标记。
- 619** 未封闭的数据部分。
- 620** 未封闭的标记；例如，文档中最后一个标记之后的文本。
- 621** 文档中存在未能解析的实体。
- 622** 意外错误。
- 631** 未能分析元标记中的字段或属性信息。标记的格式必须是 <meta name="abc" content="xyz">; 可能属性名或元标记的内容不正确。
- 632** 未能将实体变换为字符。
- 650** 不同字段定义以同一开始标记开始。
- 651** 因为一个开始标记包含另一个开始标记，所以这些标记有歧义。
- 652** 如果字段和属性使用同一开始标记，则它们必须使用同一结束标记或都不使用任何结束标记。
- 653** 文档结束时，字段仍然没有关闭。
- 654** 未对结构化格式指定任何文档模型。会将该文档作为没有字段或属性信息的纯文本文档进行分析。

- 670** 因为操作需要 “Outside In” (TM) 库, 而又找不到该库, 所以未能执行该操作。
- 671** 因为未能从 “Outside In” (TM) 库装入必需的过程, 所以未能执行该操作。这些库可能过期或已损坏。
- 672** 使用 “Outside In” 处理文档时发生错误。

附录 J. 故障诊断

DB2 Net Search Extender 提供下列信息以跟踪故障。

跟踪故障

如果需要向 IBM 代表报告错误，则可能会要求您打开跟踪以便可将信息写至可用于查找错误的文件。

因为跟踪打开时系统性能会受影响，所以仅在“IBM 支持中心”代表或您的技术支持代表指导下才应使用跟踪工具。

要打开跟踪，使用 DB2 工具：

```
db2trc on
```

有关进一步的信息，参见 *DB2 UDB 命令参考文档*。

要接收特定于 Net Search Extender 的信息，可使用带有组件 96 的屏蔽：

```
db2trc on -m *.*.96.*.*
```

在发生严重错误的情况下，查看 db2diag.log 可能也有帮助。

附录 K. Data Link 消息

Data Link 中的错误不会导致 Net Search Extender 错误，并从而导致建立索引过程终止。确保检查事件日志并注意未建立索引的文档。需要手工对这些文档重新建立索引（如果需要的话）。

表 17. Data Link 警告消息

编号	Data Link 消息
01H90=	CTEDL — 设置返回 BLOB 值时出错。
01H91=	CTEDL — DataLink I/O 操作超时。
01H92=	CTEDL — 字符编码不受支持。
01H93=	CTEDL — 不受支持的 DataLink 方案。
01H94=	CTEDL — 创建数据类型 BLOB 的实例时出错。
01H95=	CTEDL — 设置代理信息时出错。
01H96=	CTEDL — UNC 方案仅在 Windows 上有效，但未能确定 OS。
01H97=	CTEDL — UNC 方案仅在 Windows OS 上有效。
01H98=	CTEDL — DFS 方案仅在 AIX 上有效。
01H99=	CTEDL — DFS 方案仅在 AIX 上有效，但是未能确定 OS。
01H01=	CTEDL — 确定 url 字符串中的端口号时出错。
01H02=	CTEDL — 检测到未知 Datalink 方案。
01H03=	CTEDL — 未能建立连接。
01H00=	CTEDL — 执行 DataLink UDF 期间出错。
01H80=	CTEDL — 找不到 DataLink 文件 "{0}"。
01H81=	CTEDL — 未授权的对 "{0}" 的 DataLink 文件存取。
01H82=	CTEDL — 文件意外结束或到达 "{0}" 的流的末尾。
01H83=	CTEDL — DataLink 文件 "{0}" 不可读。
01H85=	CTEDL — DataLink URL 方案 "{0}" 需要文件名。
01H86=	CTEDL — 未建立与 DataLink 文件服务器 "{0}" 的任何连接。
01H60=	CTEDL — 错误的 HTTP 请求 — 变形的 DataLink URL 语法。
01H61=	CTEDL — 未授权的 DataLink 请求 — 需要用户认证。
01H62=	CTEDL — DataLink 存取需要付费。
01H63=	CTEDL — 禁止存取 DataLink URL。
01H64=	CTEDL — 在 DataLink 服务器上找不到文件。

表 17. Data Link 警告消息 (续)

01H65=	CTEDL — 对于 DataLink 资源, 不允许请求的方法。
01H66=	CTEDL — 请求不可接受。
01H67=	CTEDL — 需要代理认证。
01H68=	CTEDL — 客户机请求超时。
01H69=	CTEDL — 与 DataLink 资源的当前状态冲突。
01H10=	CTEDL — DataLink 资源在服务器上不再可用。
01H11=	CTEDL — 必须指定内容长度才能接受请求。
01H12=	CTEDL — 标题字段中给定的前置条件求值为 false。
01H13=	CTEDL — 请求的 DataLink 实体过大。
01H14=	CTEDL — 请求的 DataLink URL 过长。
01H15=	CTEDL — 不受支持的 MIME 类型。
01H16=	CTEDL — 不能满足范围请求。
01H17=	CTEDL — 期望失败。
01H18=	CTEDL — 对 DataLink URL 的请求未获取任何内容长度信息。
01H20=	CTEDL — HTTP 响应无效。
01H70=	CTEDL — 内部 DataLink 服务器错误。
01H71=	CTEDL — DataLink 服务器不支持此功能。
01H72=	CTEDL — 错误网关。
01H73=	CTEDL — 服务不可用 — DL 服务器临时过载或维护。
01H74=	CTEDL — 网关超时。
01H75=	CTEDL — HTTP 版本不受支持。
01H30=	CTEDL — 尝试建立套接字连接 — 基本协议出错。
01H31=	CTEDL — 未能建立至 DataLink 服务器 {0} 的路由。
01H32=	CTEDL — 未能将套接字连接至远程地址 {0}。
01H33=	CTEDL — 未能将套接字绑定至本地地址。
01H34=	CTEDL — 未能确定 DataLink 服务器 {0} 的 IP 地址。
01H35=	CTEDL — 未知服务异常 — 没有任何 MIME 类型支持。
01H36=	CTEDL — 变形的 URL '{0}' — 没有任何受支持的协议或未能分析 DL URL。

附录 L. 同义字支持的 CCSID

同义字支持下列 CCSID:

CCSID

819	拉丁语 1
850	PC 数据拉丁语 1
874	泰国语
932	组合的日语
943	组合的日语
950	组合的繁体中文
954	日语
970	组合的韩国语
1208	UTF 8
1250	拉丁语 2
1252	拉丁语 1
1253	捷克语
1254	土耳其语
1255	希伯来语
1256	阿拉伯语
1258	越南语
1363	组合的韩国语
1381	组合的简体中文
1383	中文（简体），组合的 SBCS/DBCS
1386	中文（简体），组合的 SBCS/DBCS
5039	日语（组合的 SBCS/DBCS）

要编译同义字定义文件，参见第 149 页的『DB2EXTTH（实用程序）』。

附录 M. 同义字工具返回的消息

ADM_MSG_INVALID_CCSID

解释: 指定了无效的 CCSID。
请求的代码页不受支持。

ITL_THES_MSG_DEFFILE_MISSING

解释: 参数错误文件名。同义字定义文件不存在。

ITL_THES_MSG_NONAME_ERROR

解释: 参数错误。未指定任何同义字定义文件名。

ITL_THES_MSG_PATHLEN_ERROR

解释: 参数错误文件名。同义字定义文件路径过长。路径长度一定不能超出操作系统中的目录名支持的最大长度。

ITL_THES_MSG_NAMELEN_ERROR

解释: 参数错误文件名。同义字定义文件名过长。

ITL_THES_MSG_NO_TARGET_DIR_ERROR

解释: 参数错误。未指定任何目标目录。

ITL_THES_MSG_UNEXPECTED_ERROR

解释: 出现内部意外错误。

ITL_THES_MSG_PARAMETER_ERROR

解释: 内部参数错误。

ITL_THES_MSG_FILE_OPEN_ERROR

解释: 打不开文件文件名。

ITL_THES_MSG_FILE_REACHED_END

解释: 同义字定义文件中出现文件意外结束。
定义文件中存在错误。

ITL_THES_MSG_FILE_READ_ERROR

解释: 未能读取文件文件名。

ITL_THES_MSG_FILE_WRITE_ERROR

解释: 未能写文件文件名。

ITL_THES_MSG_FILE_ACCESS_ERROR

解释: 未能存取文件文件名。

ITL_THES_MSG_FILE_REMOVE_ERROR

解释: 未能除去文件文件名。

ITL_THES_MSG_FILE_RENAME_ERROR

解释: 未能将文件文件名 1 重命名为文件名 2。

ITL_THES_MSG_FILE_CLOSE_ERROR

解释: 未能关闭文件文件名。

ITL_THES_MSG_FILE_EOF_ERROR

解释: 文件名中出现文件意外结束。
定义文件中存在错误。

ITL_THES_MSG_MEMORY_ERROR

解释: 内存错误。

ITL_THES_MSG_BUFFER_OVERFLOW

解释: 缓冲区溢出。

ITL_THES_MSG_LOCKING_ERROR

解释: 未能锁定字典文件名。

ITL_THES_MSG_LOCKED

解释: 正在使用同义字典字典名。

ITL_THES_MSG_OUTFILE_EXIST

解释: 输出文件文件名已存在。

ITL_THES_MSG_DICT_INTEGRITY_ERROR

解释: 字典字典名不完整。

同义字典文件已损坏。

ITL_THES_MSG_DICT_VERSION_ERROR

解释: 字典字典名版本错误。

同义字典是使用不兼容的早期版本创建的。

ITL_THES_MSG_DICT_NOT_EXIST

解释: 同义字典字典名不存在。

ITL_THES_MSG_DICT_EXIST

解释: 同义字典字典名已存在。

不能覆盖。

ITL_THES_MSG_NORMALIZE_ERROR

解释: 规范化术语时出错。

同义字定义文件中存在错误。

ITL_THES_MSG_INTERNAL_ERROR

解释: 内部错误。

ITL_THES_MSG_INPUT_ERROR

解释: 同义字定义文件文件名的第行号行存在错误。

ITL_THES_MSG_ERROR_IN_FILE

解释: 文件文件名中存在错误。

ITL_THES_MSG_IE_EMPTY

解释: 同义字定义文件文件名是空的。

ITL_THES_MSG_IE_BLOCK_START

解释: 在文件文件名的第行号行中找不到任何块起始行。

ITL_THES_MSG_IE_REL_SYNTAX

解释: 在文件名的第行号行中不正确地指定了关系。

ITL_THES_MSG_IE_USER_DEF

解释: 在文件名的第行号行中不正确地指定了关系。

ITL_THES_MSG_IE_USER_DEF_DOMAIN

解释: 关系编号超出文件名第行号行的范围。

ITL_THES_MSG_IE_NO_TERM

解释: 在文件名的第行号行中未定义任何术语。

ITL_THES_MSG_IE_TERM_LEN

解释: 同义字术语长度超过 64 个字符。

ITL_THES_MSG_IE_STRENGTH_SYNTAX

解释: 不正确地指定了强度值。

语法: 在该术语后输入 [:20] 表示强度为 20。

ITL_THES_MSG_IE_STRENGTH_DOMAIN

解释: 强度超出范围。

有效值为 1 - 100; 缺省值为 100。

附录 N. Windows 系统错误

以下是 Windows 系统错误的列表:

系统错误

1	不正确的函数。
2	系统找不到指定的文件。
3	系统找不到指定的路径。
4	系统打不开文件。
5	拒绝访问。
6	句柄无效。
8	没有足够的存储空间可用来处理此命令。
14	没有足够的存储空间可用来完成此操作。
15	系统找不到指定的驱动器。
29	系统不能写入指定的设备。
30	系统不能从指定的设备读取。
32	进程不能存取该文件，原因是它正由另一个进程使用。
36	打开太多的文件进行共享。
38	已到达文件的末尾。
39	磁盘已满。
80	文件已存在。
82	不能创建目录或文件。
100	不能创建另一个系统信号。
101	专用信号由另一个进程所拥有。
102	设置了信号并且不能关闭它。
103	不能再次设置该信号。
104	不能在中断时请求专用信号。
105	此信号的前一个所有权已结束。

110	系统打不开指定的设备或文件。
111	文件名太长。
112	磁盘上没有足够空间。
121	信号超时这一时间段已到期。
126	找不到指定的模块。
127	找不到指定的过程。
147	没有足够的资源可用来处理此命令。
155	不能创建另一线程。
161	指定的路径无效。
164	不能在系统中再创建其它线程。
170	请求的资源在使用中。
183	当文件已存在时不能创建该文件。
187	找不到指定的系统信号名称。
206	文件名或扩展名太长。
267	目录名无效。
288	尝试释放并非由调用者所拥有的互斥。
298	对信号提出了太多公布要求。
998	对内存位置的存取无效。
1051	停止控制权已发送至其它运行的服务所依赖的服务。
1052	请求的控制权对此服务无效。
1053	服务未及时响应启动或控制请求。
1054	未能为服务创建线程。
1055	服务数据库已锁定。
1056	服务的实例已在运行。
1057	帐户名无效或不存在。
1058	不能启动服务，原因是已禁用该服务或它没有与服务相关联的已启用设备。
1059	指定了循环服务相关性。
1060	指定的服务不是作为已安装的服务存在。

1061	此时服务不能接受控制消息。
1062	服务尚未启动。
1063	服务进程未能连接至服务控制器。
1064	处理控制请求时服务中发生了异常。
1066	服务已返回特定于服务的错误代码。
1067	进程意外终止。
1068	相关性服务或组未能启动。
1069	因为登录失败，所以服务未启动。
1070	启动之后，服务处于启动暂挂状态。
1071	指定的服务数据库锁定无效。
1072	指定的服务已标记为删除。
1073	指定的服务已存在。
1078	该名称已用作服务名或服务显示名。
1079	为此服务指定的帐户不同于为在同一进程中运行的其它服务指定的帐户。
1082	未对此服务配置任何恢复程序。
1154	运行此应用程序所需的一个库文件已损坏。
1219	提供的凭证与现有的一组凭证冲突。
1242	已注册该服务。
1243	指定的服务不存在。
1244	未执行正在请求的操作，原因是尚未认证用户。
1245	未执行正在请求的操作，原因是用户尚未登录网络。指定的服务不存在。
1392	文件或目录已损坏且不是可读的。
1455	页面调度文件太小，此操作无法完成。
1793	用户帐户已到期。

附录 O. 声明

IBM 可能在其它国家或地区不提供本文档中讨论的产品、服务或功能特性。有关您当前所在区域的产品和服务的信息，请向您当地的 IBM 代表咨询。任何对 IBM 产品、程序或服务的引用并非意在明示或暗示只能使用 IBM 的产品、程序或服务。只要不侵犯 IBM 的知识产权，任何同等功能的产品、程序或服务，都可以代替 IBM 产品、程序或服务。但是，评估和验证任何非 IBM 产品、程序或服务，则由用户自行负责。

IBM 公司可能已拥有或正在申请与本文档内容有关的各项专利。提供本文档并未授予用户使用这些专利的任何许可证。您可以用书面方式将许可证查询寄往：

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

关于双字节 (DBCS) 许可查询的信息，请与您所在国家或地区的“IBM 知识产权部门”联系，将查询以书面方式寄往：

IBM World Trade Asia Corporation
Licensing
2-31 Roppongi 3-chome, Minato-ku
Tokyo 106, Japan

本条款不适用于英国或任何这样的条款与当地法律不一致的国家或地区：国际商业机器公司以“按现状”的基础提供本出版物，不附有任何形式的（无论是明示的，还是默示的）保证，包括（但不限于）对非侵权性、适销性和适用于某特定用途的默示保证。某些国家或地区在某些交易中不允许免除明示或默示的保证，因此，本条款可能不适用于您。

本信息中可能包含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；这些更改将编入本资料的新版本中。IBM 可以随时对本资料中描述的产品和 / 或程序进行改进和 / 或更改，而不另行通知。

本信息中对非 IBM Web 站点的任何引用都只是为了方便起见才提供的，不以任何方式充当对那些 Web 站点的保证。该 Web 站点中的资料不是 IBM 产品资料的一部分，使用那些 Web 站点带来的风险将由您自行承担。

IBM 可以它认为合适的任何方式使用或分发您所提供的任何信息，而无须对您承担任何责任。

本程序的被许可方如果要了解有关程序的信息以达到如下目的：（i）允许在独立创建的程序和其它程序（包括本程序）之间进行信息交换，以及（ii）允许对已经交换的信息进行相互使用，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

只要遵守适当的条件和条款，包括某些情形下的一定数量的付费，都可获得这方面的信息。

本资料中描述的许可程序及其所有可用的许可资料均由 IBM 依据 IBM 客户协议、IBM 国际程序许可证协议或任何同等协议中的条款提供。

此处包含的任何性能数据都是在受控环境中测得的。因此，在其它操作环境中获得的数据可能会有明显的不同。有些测量可能是在开发级的系统上进行的，因此不保证与一般可用系统上进行的测量结果相同。此外，有些测量是通过推算而估计的，实际结果可能会有差异。本文档的用户应当验证其特定环境的适用数据。

涉及非 IBM 产品的信息可从这些产品的供应商、其出版说明或其它可公开获得的资料中获取。IBM 没有对这些产品进行测试，也无法确认其性能的精确性、兼容性或任何其它关于非 IBM 产品的声明。有关非 IBM 产品性能的问题应当向这些产品的供应商提出。

所有关于 IBM 未来方向或意向的声明都可随时更改或收回，而不另行通知，它们仅仅表示了目标和意愿而已。

本资料中可能包含用于日常业务运作的数据和报表的示例。为了尽可能完整地说明问题，这些示例可能包含个人、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有雷同，纯属巧合。

版权许可证：

本资料中可能包含源语言的样本应用程序，它们举例说明了各种操作平台上的编程技术。为了开发、使用、营销或分发符合编写这些样本程序所针对操作平台的应用程序编程接口的应用程序，您可以以任何形式复制、修改和分发这些样本程

序，而不必向 IBM 付款。尚未在所有条件下彻底测试这些示例。因此，IBM 不能保证或默示这些程序的可靠性、适用性或功能。

这些样本程序或任何派生产品的每个副本或任何部分都必须包括如下版权声明：

©（您的公司名称）（年度）。本代码的某些部分是从“IBM 公司样本程序”派生的。© Copyright IBM Corp. _输入年份_. All rights reserved.

商标

下列各项是国际商业机器公司在美国和 / 或其它国家或地区的商标：

AIX	DB2 Universal Database
DB2	IBM DRDA
DB2 Extender	z/OS
Informix	

下列各项是其它公司的商标或注册商标：

Microsoft、Windows、Windows NT 和 Windows 徽标是 Microsoft Corporation 在美国和 / 或其它国家或地区的商标。

Java 和所有基于 Java 的商标是 Sun Microsystems, Inc. 在美国和 / 或其它国家或地区的商标。

Intel、Intel Inside（徽标）、MMX 和 Pentium 是 Intel Corporation 在美国和 / 或其它国家或地区的商标。

UNIX 是 The Open Group 在美国和其它国家或地区的注册商标。

其它公司、产品和服务名称可能是其它公司的商标或服务标记。

词汇表

本词汇表定义本手册中所使用的术语和缩写词。如果找不到您要查找的术语，则参考索引或者 *Dictionary of Computing*, New York: McGraw-Hill, 1994。

A

存取函数 (access function)： 用户提供的函数，它将一列中存储的文本的数据类型转换为可以由 DB2 Net Search Extender 处理的类型。

B

布尔搜索 (Boolean search)： 一种搜索方法，在该搜索方法中，将使用布尔运算符来组合一个或多个搜索术语。

C

目录视图 (catalog view)： DB2 Net Search Extender Text 为了便于管理而创建的系统表的视图。目录视图包含有关已启用以供 DB2 Net Search Extender Text 使用的表和列的信息。

CCSID： 编码字符集标识符。

代码页 (code page)： 将图形字符和控制功能含义指定给所有代码点。例如，对于 8 位代码，将字符和含义指定给 256 个代码点。

命令行处理器 (command line processor)： 一个称为 db2text 的程序，它可以：

- 允许您输入 DB2 Net Search Extender 命令
- 处理命令
- 显示结果

计数 (count)： 一个关键字，用来指定同义字中术语的层数（深度），这些同义字将用来扩展给定关系的搜索术语。

D

DBCS： 双字节字符集。

禁用 (disable)： 通过除去在启用过程中创建的项，将数据库复原为 DB2 Net Search Extender Text 启用之前的状态。

文档 (document)： 参见文本文档 (text document)。

文档模型 (document model)： 用文档包含的各节来表示的文档结构的定义。文档模型使 DB2 Net Search Extender 在建立索引时能够了解文档中的各节。文档模型列示用来标识各节的标记 (markup tag)。对于每个标记，可以指定一个描述性节名，以便在对该节进行查询时使用。可以在文档模型文件中指定一个或多个文档模型。

E

启用 (enable)： 准备数据库以供 DB2 Net Search Extender 使用。

转义字符 (escape character)： 用来指示不要将后续字符解释为屏蔽字符 (masking character) 的字符。

扩展 (expand)： 它是将从同义字派生的附加术语添加至搜索术语的一种操作。

F

格式 (format)： 文档的类型，例如，ASCII 或 HTML。

自由文本搜索 (free-text search)： 一种搜索方法，在该搜索方法中，搜索术语是以自由格式的文本（用自然语言描述要搜索的主题的短语或句子）表示的。

函数 (function)： 参见 *存取函数 (access function)*。

模糊搜索 (fuzzy search)： 一种搜索方法，它可以找到其拼写与搜索术语的拼写相似的词语。

H

混合搜索 (hybrid search)： 它将布尔搜索 (*Boolean search*) 和自由文本搜索 (*free-text search*) 组合起来使用。

I

索引 (index)： 从文本中抽取重要术语，并将它们存储在文本索引 (*text index*) 中。

索引特征 (index characteristics)： 用来确定以下内容的文本索引 (*text index*) 的属性：

- 更新索引的频率

- 发生第一次索引更新的时间

L

日志表 (log table)： 由 DB2 Net Search Extender 创建的一个表，它包含有关要对哪些文本文档建立索引的信息。每当添加、更改或删除已启用的文本列中的文档时，就使用触发器 (*Trigger*) 来将此信息存储在日志表中。

M

屏蔽字符 (masking character)： 在搜索术语前面、中间或末尾用来表示可选字符的字符。屏蔽字符通常用于在精确索引中查找术语的变体。

匹配 (match)： 搜索术语在文本文档中的出现。

P

定期建立索引 (periodic indexing)： 按预先确定的时间间隔建立索引，时间间隔是用天、小时、分钟以及在可建立索引之前日志表 (*log table*) 中应当列出的最小文档名称数来表示。

R

检索 (retrieve)： 在 DB2 Net Search Extender 的其中一个搜索函数中使用搜索自变量来查找文本文档。

S

SBCS： 单字节字符集。

分数 (Score)： 从 0 到 1 之间的、类型为 DOUBLE 的一个绝对值，它指示一个文档相对于找到的其它文档来说满足搜索条件的程度。该值指示在文档中找到的匹配数与文档大小的比例。该值指示在文档中找到的匹配数与文档大小成比例。

搜索自变量 (search argument)： 在进行搜索时所指定的条件，它由一个或几个搜索术语以及搜索参数组成。

T

文本列 (text column)： 包含文本文档 (*text document*) 的列。

文本文档 (text document)： 存储在 DB2 表中的类型为

CHAR、GRAPHIC、VARGRAPHIC、LONG VARGRAPHIC、DBCLOB、VARCHAR、LONG VARCHAR 或 CLOB 数据类型的文本。

文本索引 (text index)： 从文本文档中抽取的重要术语的集合。每个术语都与抽取该术语的文档相关联。通过在索引中而不是在文档本身中进行搜索，明显缩短了搜索时间。

跟踪 (tracing) : 它是这样一种操作: 将信息存储在文件中, 然后可以使用这些信息来查明错误原因。

触发器 (trigger) : 触发器是以下一种机制: 每当对文本列添加、更改或删除文档时, 自动地将有关需要建立索引的文档的信息添加到日志表 (*log table*) 中。

U

UDF: 用户定义函数。

UDT: 用户定义类型。

更新频率 (update frequency) : 更新文本索引的频率, 是用天、小时、分钟以及在可建立索引之前日志表 (*log table*) 中应当列出的最小文档名称数来表示。

用户定义类型 (user-defined type) (UDT) : 由 DB2 的用户创建的一种数据类型, 与 DB2 提供的数据类型 (例如, LONG VARCHAR) 相对。

用户定义函数 (user-defined function) (UDF) : 由 DB2 的用户创建的 SQL 函数, 与 DB2 提供的 SQL 函数相对。

W

通配符 (wildcard character) : 参见屏蔽字符 (*masking character*) 。

索引

[A]

安装 13
安装验证 16
安装 Data Links jar 文件 43

[B]

备份与复原索引 55
表空间 40, 63
表值函数
 使用复制对呢称创建文本索引 43
 有关视图的文本索引 48
表值搜索函数
 使用 HIGHLIGHT 函数 92
 搜索 92
 语法 171
 HIGHLIGHT 语法 175
布尔运算符
 搜索语法 162
 NOT 89
 & (AND) 和 | (OR) 86

[C]

重新创建索引 51
触发器
 创建 40
 描述 5
存储过程
 概述 44
 更新文本索引 47
 激活文本索引 47
 取消激活文本索引 47
 使用“DB2 复制”来对呢称创建
 文本索引 43
 有关视图的文本索引 48
存储过程函数
 搜索 91
 语法 180

[F]

分区 DB2 服务器的安装 14
服务器
 跟踪故障 271
 启动 110
 停止 111
附加概念 8
附录
 标记化 261
 故障诊断 271
 迁移 203
 使用大量内存 207
 受支持的语言 229
 数据链路消息 273
 同义字工具返回的消息 277
 同义字支持的 CCSID 275
 文本搜索引擎 261
 文本搜索引擎原因代码 263
 文档模型 253
 无用词 261
 信息目录 211
 CCSID 221
 Net Search Extender 消息 233
 Windows 系统错误 281
复制捕获表 137, 218

[G]

概念
 附加概念 8
 关键 3
 列变换函数 9
 实例服务 9
 使用表值函数 8
 使用存储过程搜索 7
 使用 SQL 标量搜索函数 6
 视图 9
概述 3
跟踪故障 271
更改数据链路返回大小 59

更改 DATALINK 返回大小 43
更新服务 32
更新频率 51
故障查找 271
故障诊断 271
关键概念 3
关键功能 10
关键术语 3
管理

 备份与复原索引 55
 查看文本索引状态 54
 创建文本索引 39, 61
 改变文本索引设置 52, 74
 跟踪故障 271
 更新文本索引 51, 77
 激活高速缓存 79
 启动 DB2 Net Search
 Extender 29, 58, 110
 清除索引事件 53, 78
 取消激活高速缓存 79
 删除文本索引 54, 77
 实例所有者命令总结 107
 使用锁定服务 30
 数据库管理员命令摘要 113
 停止 DB2 Net Search
 Extender 29, 58, 111
 维护文本索引 50, 73
 文本表所有者命令总结 121
 显示索引状态 80
 DB2 控制中心 57
规划 25

[H]

函数
 参考 167
 存储过程 180
 概述 167
 描述 83
 搜索函数 83
 搜索文本 84

函数 (续)

- 用于转换数据类型 41
- 指定搜索自变量 85
- CONTAINS 168
- HIGHLIGHT 175
- NUMBEROFMATCHES 169
- SCORE 170
- SQL 表值 171
- 环境, 客户机 / 服务器 11

[J]

结构化文档

- 概述 97
- 启用节支持 185
- 缺省文档模型 185
- 示例 88
- 搜索语法 163

[K]

- 客户机 / 服务器环境 11
- 扩展搜索术语
 - 请看 同义字

[L]

- 列变换函数 9

[M]

面板和对话框

- “改变索引”对话框 74
- “高速缓存表”面板 68
- “更新索引”对话框 77
- “更新特征”面板 67
- “激活高速缓存”对话框 79
- “名称”面板 61
- “目标”面板 63
- “取消激活高速缓存表”对话框 79
- “删除索引”对话框 77
- “索引事件”对话框 78
- “索引状态”对话框 80
- “文本属性”面板 65

面板和对话框 (续)

- “总结”面板 72

命令

- ACTIVATE CACHE 122
- ALTER INDEX 124
- CLEAR EVENTS 128
- CONTROL 108
- COPYRIGHT 157
- CREATE INDEX 130
- DB2EXTDL 118
- DB2EXTHL 119
- DB2EXTTH 149
- db2text 107, 113, 121
- DEACTIVATE CACHE 145
- DISABLE DATABASE 116
- DROP INDEX 147
- ENABLE DATABASE 114
- HELP 155
- START 110
- STOP 111
- UPDATE INDEX 151

命令摘要

- 适用于数据库管理员 113
- 文本表所有者的 121
- 用于实例所有者 107

模糊搜索, 示例 87

- 目录名和文件名 15
- 目录视图 37

[N]

- 内存量 207

[P]

匹配

- 在搜索结果中 84
- NUMBEROFMATCHES 函数 169
- 平面 ASCII, 文档格式 25

[Q]

- 启动 DB2 Net Search Extender 110
- 迁移 203
- 缺省文档模型 185

[R]

日志表

- 创建 40
- 描述 5

日志表视图 218

- 入门 19

[S]

识别

- 词语 261
- 段落 261
- 句子 261
- 无用词 261

实例服务 9, 29, 107

- 使用大量内存 207
 - 对于 AIX 207
 - 对于 Sun Solaris 208
 - 对于 Windows 207
- HP-UX 209
- Linux 209

事件视图 217

视图

- 复制捕获表 218
- 概述 9
- 日志表视图 218
- 事件视图 217
- 已创建的视图 37
- db2ext.dbdefaults 37, 211
- db2ext.indexconfiguration 37, 215
- db2ext.proxyinformation 37, 213
- db2ext.textindexes 37, 54, 214
- db2ext.textindexformats 37, 216

受支持的语言 229

数据

- 外部存储的 9

数据库

- 备份与复原索引 55
- 禁用数据库 38, 59
- 启用数据库 37, 59

搜索多列, 示例 94

搜索函数

- 存储过程 180
- CONTAINS 168
- HIGHLIGHT 175

搜索函数 (续)

NUMBEROFMATCHES 169

SCORE 170

SQL 表值 171

搜索术语的出现次数 169

搜索术语扩展

请看 同义字

搜索术语中的屏蔽字符 87

搜索术语中的通配符 87

搜索文本

概述 84

获取找到的匹配项数目 84

获取找到的文档的分数 85

进行查询 84

使用表值函数 92

使用存储过程 91

使用 HIGHLIGHT 函数 92

语法 160

搜索自变量

按固定顺序搜索术语 88

按任意顺序搜索术语 86

布尔运算符 162

描述 159

模糊搜索 87, 164

使用屏蔽字符 87

使用通配符 87

使用 NOT 进行搜索 89

使用 & 和 | 进行搜索 86

属性名称 165

数字属性搜索 89

搜索术语的一部分 87

同义字搜索 89

语法 160

在同一段落中搜索术语 88

在同一句子中搜索术语 88

在文档的各节中搜索术语 88

指定 85

自由文本搜索 90

search-primary 运算符 163

搜索自变量关键字

COUNT 165

EXPAND 165

EXPANSION LIMIT 162

FUZZY FORM OF 164

PRECISE FORM OF 164

RESULT LIMIT 162

搜索自变量关键字 (续)

SECTION 163

STEMMED FORM OF 164

STOP SEARCH AFTER number

DOCUMENTS(S) 162

TERM OF 165

THESAURUS 165

搜索自变量中的 & (AND) 运算符

如何使用 86

搜索自变量中的 | (OR) 运算符

如何使用 86

锁定服务

查看 32

使用 29

CONTROL 命令 108

索引

备份与复原 55

查看文本索引状态 54

大小计算 25

改变文本索引设置 52, 74

概述 3

更新频率 51

更新文本索引 51, 77

关系 100

规划 25

激活高速缓存 79

清除索引事件 53, 78

取消激活高速缓存 79

删除文本索引 54, 77

使用结构化文档 97

维护文本索引 50, 73, 99

显示索引状态 80

DB2 控制中心 57

索引的空间需求 25

索引更新事件

记录 40

删除 53

[T]

停止 DB2 Net Search Extender 111

通用 (GPP) 文档

定义文档模型 186

文档格式 25

文档模型的文档类型定义 253

限制 256

同义字

编译 101

创建 101

定义文件 101

概念 99

结构 99

受支持的 CCSID 275

同义字定义语法 197

消息 277

同义字编译实用程序 149

同义字搜索

示例 89

语法 165

THESAURUS 关键字 165

同义字中的关系 100

同义字中术语的深度, 指定 165

[W]

外部存储的数据 9

文本搜索引擎

标记化 261

无用词 261

语言支持无用词 262

原因代码 263

文本特征

格式 25

CCSID 25

文本文档的格式 25

描述 25

受支持的列表 25

文档

格式, 描述 25

建立索引 3

结构 185

受支持的格式 25

转换数据类型 41

CCSID 25

文档类型 25

文档模型

概述 97

描述 185

缺省 185

搜索语法中的属性名 165

搜索语法中的 SECTION 关键字
163

文档模型 (续)

文档类型定义 253

限制 256

修改 185

文档模型参考 253

文档数据类型

二进制数据类型 41

转换不受支持的数据类型 41

DATALINK 数据类型 42

无用词 261

[X]

系统需求 13

信息目录 211

性能注意事项

对于建立索引 50

对于搜索 95

[Y]

样本函数

运行 90

用户方案

存储过程搜索示例 21

SQL 标量搜索示例 19

SQL 表值函数示例 22

用于索引的磁盘空间 25

[Z]

主键类型 144

转义字符

使用 88

[特别字符]

“改变索引”对话框 74

“高速缓存表”面板 68

“更新索引”对话框 77

“更新特征”面板 67

“激活高速缓存”对话框 79

“名称”面板 61

“目标”面板 63

“取消激活高速缓存表”对话框 79

“删除索引”对话框 77

“索引事件”对话框 78

“索引状态”对话框 80

“文本属性”面板 65

“总结”面板 72

A

ACTIVATE CACHE 命令

使用 47

语法 122

AIX 安装 14

ALTER INDEX 命令

使用 52

语法 124

ASCII, 文档格式 25

C

CCSID

列表 221

同义字支持的 275

文档代码页 25

CLEAR EVENTS 命令

使用 53

语法 128

COMMITCOUNT

关键字 139

性能注意事项 50

CONTAINS 函数

示例 84

语法 168

CONTROL 命令

使用 32

语法 108

COPYRIGHT 命令

语法 157

COUNT 关键字 165

CREATE INDEX 命令

使用 39

语法 130

D

Datalink Manager

安装 jar 文件 43

Datalink Manager (续)

错误消息 273

更改数据链路返回大小 118

更改 DATALINK 返回大小 43

DATALINK 数据类型 42

DB2 控制中心

创建文本索引 61

管理 57

禁用数据库 59

启动和停止 DB2 Net Search

Extender 58

启用数据库 59

使用向导 61

维护文本索引 73

“改变索引”对话框 74

“高速缓存表”面板 68

“更新索引”对话框 77

“更新特征”面板 67

“激活高速缓存”对话框 79

“名称”面板 61

“目标”面板 63

“取消激活高速缓存表”对话框

79

“删除索引”对话框 77

“索引事件”对话框 78

“索引状态”对话框 80

“文本属性”面板 65

“总结”面板 72

DB2 Net Search Extender 概述 3

DB2EXTDL 命令

使用 149

语法 118

DB2EXTHL 命令

语法 119

db2ext.dbdefaults 视图 211

db2ext.indexconfiguration 视图 215

db2ext.proxyinformation 视图 213

db2ext.textindexformats 视图 216

DB2TX, 命令行处理器

使用 35

语法 107, 113, 121

DEACTIVATE CACHE 命令

使用 47

语法 145

DISABLE DATABASE 命令

使用 38, 59

DISABLE DATABASE 命令 (续)

语法 116

DROP INDEX 命令

使用 54

语法 147

E

ENABLE DATABASE 命令

使用 37, 59

语法 114

EXPAND 关键字 165

EXPANSION LIMIT 关键字 162

F

FUZZY FORM OF 关键字 164

H

HELP 命令

语法 155

HIGHLIGHT 函数

更改 CLOB 大小 119

使用 TEXTSEARCH 函数 175

示例 176

语法 175

HTML 文档

定义文档模型 189

结构化文档 185

缺省文档模型 185

文档格式 25

文档模型的文档类型定义 253

限制 256

I

INSO 格式

参见 Outside-In 过滤软件 16

N

Net Search Extender

备份与复原 55

查看文本索引状态 54

Net Search Extender (续)

创建文本索引 39, 61

创建用于存储过程搜索的高速缓存
44

改变文本索引设置 52, 74

更新服务 32

更新文本索引 51, 77

激活高速缓存 79

禁用数据库 38, 59

启动和停止 29, 58

启用数据库 37, 59

清除索引事件 53, 78

取消激活高速缓存 79

删除文本索引 54, 77

实例服务 29

使用 DB2 控制中心 57

使用“DB2 复制”来对昵称创建

文本索引 43

维护文本索引 50, 73

显示索引状态 80

消息 233

Net Search Extender 信息目录

查看视图 211

NUMBEROFMATCHES 函数

示例 84

语法 169

O

OR 布尔运算符 86

Outside-In 过滤软件

安装库 16

标记属性 257

定义文档模型 195

简介 26

结构化文档 185

缺省文档模型 185

文档格式 25

P

PRECISE FORM OF 关键字 164

R

RESULT LIMIT 关键字 162

S

SCORE 函数

示例 85

语法 170

search-primary 运算符 163

START 命令

使用 29, 58

语法 110

STEMMED FORM OF 关键字 164

STOP 命令

使用 29, 58

语法 111

STOP SEARCH AFTER number

DOCUMENTS(S) 关键字 162

T

TERM OF 关键字 165

TEXTSEARCH 函数

使用 HIGHLIGHT 函数 171

示例 173

语法 171

U

UNIX 安装 14

UNIX 安装验证 16

UPDATE INDEX 命令

对于存储过程 47

更新频率 51

使用 52

语法 151

RECREATE 选项 51

user 角色

数据库管理员 28

文本表所有者 28

DB2 实例所有者 27

W

Windows 安装 15

Windows 安装验证 16

Windows 系统错误 281

X

XML 文档

定义文档模型 191

结构化文档 185

缺省文档模型 185

文档格式 25

文档模型的文档类型定义 253

限制 256

XPath 表达式语义 254

XPath 表达式语义 254

读者意见表

IBM DB2 通用数据库
Net Search Extender
管理和用户指南
版本 8.1

S152-0596-00

姓名

地址

单位及部门

电话号码



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 中国公司上海分公司，汉化部
中国上海市淮海中路 333 号瑞安广场 10 楼
邮政编码：200021

折起并封口

请勿使用钉书机

折起并封口

请沿此线
撕下或折起



部件号: CT202SC

中国印刷

S152-0596-00



(1P) P/N: CT202SC

