

SYBASE®

Enhanced Full-Text Search Specialty Data Store
用户指南

Adaptive Server® Enterprise

15.0.2

文档 ID: DC33113-01-1502-01

最后修订日期: 2009 年 2 月

版权所有 © 2010 Sybase, Inc. 保留所有权利。

本出版物适用于 Sybase 软件 and 任何后续版本, 除非在新版本或技术声明中另有说明。此文档中的信息如有更改, 恕不另行通知。此处说明的软件按许可协议提供, 其使用和复制必须符合该协议的条款。

若要订购附加文档, 美国和加拿大的客户请拨打客户服务部门电话 (800) 685-8225 或发传真至 (617) 229-9845。

持有美国许可协议的其它国家 / 地区的客户可通过上述传真号码与客户服务部门联系。所有其它国际客户请与 Sybase 子公司或当地分销商联系。仅在定期安排的软件发布日期提供升级。未经 Sybase, Inc. 的事先书面许可, 本书的任何部分不得以任何形式、任何手段 (电子的、机械的、手动、光学的或其它手段) 进行复制、传播或翻译。

Sybase 商标可在位于 <http://www.sybase.com/detail?id=1011207> 的“Sybase 商标页” (Sybase trademarks page) 处进行查看。Sybase 和列出的标记均是 Sybase, Inc. 的商标。® 表示已在美国注册。

Java 和所有基于 Java 的标记都是 Sun Microsystems, Inc. 在美国和其它国家 / 地区的商标或注册商标。

Unicode 和 Unicode 徽标是 Unicode, Inc. 的注册商标。

提到的所有其它公司名和产品名均可能是与之相关联的相应公司的商标。

Use, duplication, or disclosure by the government is subject to the restrictions set forth in subparagraph (c)(1)(ii) of DFARS 52.227-7013 for the DOD and as set forth in FAR 52.227-19(a)-(d) for civilian agencies.

Sybase, Inc., One Sybase Drive, Dublin, CA 94568.

目录

关于本手册	ix	
第 1 章	简介	1
	增强型全文搜索引擎的功能	1
	高可用性	2
第 2 章	了解增强型全文搜索引擎	3
	增强型全文搜索引擎的组件	3
	源表	3
	Verity 集合	4
	过滤器	4
	text_db 数据库	4
	index 表	5
	text_events 表	6
	组件之间的关系	7
	全文搜索的工作方式	8
第 3 章	为全文搜索配置 Adaptive Server	11
	为增强型全文搜索引擎配置 Adaptive Server	11
	启用配置参数	12
	运行 installtextserver 脚本	12
	运行 installmessages 脚本	14
	运行 installevent 脚本	14
	命名本地 Adaptive Server	16
	创建和维护文本索引	16
	为源表设置索引	17
	创建文本索引和索引表	18
	授予对文本索引代理表的权限	19
	使数据库联机以进行全文搜索	20
	将更改传播到文本索引	20
	复制文本索引	21
	示例：启用新的数据库以进行文本搜索	22
	对欧元符号进行索引	23

第 4 章	设置 Verity 功能	25
	启用按示例查询、摘要和集群	25
	编辑主 style.prm 文件	26
	编辑单个 style.prm 文件	27
	设置用作排序规范的列	28
	对包含标记的文本使用过滤器	30
	创建自定义同义词库	32
	检查缺省同义词库（可选项）	33
	创建控制文件	33
	创建同义词库	34
	用自定义同义词库替换缺省同义词库	35
	创建主题	35
	创建大纲文件	36
	创建主题组目录	37
	创建知识库映射	38
	定义知识库映射的位置	38
	对定义的主题执行查询	39
	主题疑难解答	39
第 5 章	编写全文本搜索查询	41
	全文本搜索查询的组件	41
	缺省行为	42
	索引表中的伪列	42
	使用 score 列按相关性排列搜索结果	43
	使用 sort_by 列指定排序顺序	44
	使用 summary 列摘要文档	45
	使用伪列请求集群结果集	45
	全文本搜索运算符	47
	使用 Verity 运算符时的注意事项	48
	使用 Verity 运算符	49
	运算符修饰符	57
第 6 章	系统管理	59
	在 UNIX 上启动增强型全文本搜索引擎	59
	创建 runserver 文件	59
	在 Windows NT 上启动增强型全文本搜索引擎	60
	启动增强型全文本搜索引擎作为服务	61
	关闭增强型全文本搜索引擎	62
	修改配置参数	62
	修改配置值	64
	可用的配置参数	65
	设置缺省语言	66
	设置缺省字符集	66

	对欧元符号编制索引	67
	设置缺省排序顺序	67
	设置跟踪标志	68
	设置 Open Server 跟踪标志	69
	设置区分大小写参数	69
	增强型全文本搜索引擎的备份和恢复	70
	可自定义的备份和恢复	70
	备份 Verity 集合	71
	从备份中恢复集合和文本索引	72
第 7 章	性能和调优	73
	更新现有索引	73
	提高查询性能	74
	限制行数	74
	确保查询的连接次序正确	74
	重新配置 Adaptive Server	75
	cis cursor rows	75
	cis packet size	75
	重新配置增强型全文本搜索引擎	76
	batch_size	76
	min_sessions 和 max_sessions	76
	使用 sp_text_notify	77
	配置多个增强型全文本搜索引擎	77
	启动时创建多个增强型全文本搜索引擎	77
	添加增强型全文本搜索引擎	78
	配置其它增强型全文本搜索引擎	78
	多用户	79
	文件描述符和增强型全文本搜索	79
第 8 章	Verity 主题	81
	主题的定义	81
	主题组织	82
	权值分配	82
	使用主题大纲文件	82
	准备主题	83
	设置过程	83
	主题知识库	83
	将主题组合到知识库中	84
	主题的结构	85
	顶层主题	85
	子主题	86
	实证主题	86
	主题和子主题的关系	87

最大主题数	88
主题命名问题	88
Verity 查询语言	88
查询语言摘要	89
运算符优先规则	92
主题大纲示例	93
运算符参考	94
ACCRUE 运算符	94
ALL 运算符	94
AND 运算符	94
ANY 运算符	95
CONTAINS 运算符	95
ENDS 运算符	95
= (EQUALS) 运算符	95
FILTER 运算符	96
> (GREATER THAN) 运算符	96
>= (GREATER THAN OR EQUAL TO) 运算符	96
< (LESS THAN) 运算符	96
<= (LESS THAN OR EQUAL TO) 运算符	96
IN 运算符	97
MATCHES 运算符	97
NEAR 运算符	97
NEAR/N 运算符	98
OR 运算符	98
PARAGRAPH 运算符	98
PHRASE 运算符	99
SENTENCE 运算符	99
SOUNDEX 运算符	99
STARTS 运算符	99
STEM 运算符	99
SUBSTRING 运算符	100
THESAURUS 运算符	100
TYPO/N 运算符	100
WILDCARD 运算符	100
使用特殊通配符	101
查找非字母数字字符	101
WORD 运算符	102
修饰符参考	103
CASE 修饰符	103
MANY 修饰符	103
NOT 修饰符	103
ORDER 修饰符	104

权值和文档重要性	104
主题权值	104
哪些运算符接受权值	105
权值对重要性的影响	106
分配权值	107
自动分配权值	108
分配权值的技巧	108
更改权值	109
主题得分和文档重要性	109
设计主题	111
准备主题设计	112
了解信息需求	112
了解文档	113
使用扫描数据	113
文档样本分类	113
主题设计策略	114
自上而下设计	114
自下而上设计	115
设计初始主题	115
制定主题大纲	115
自上而下主题大纲示例	116
自下而上主题大纲示例	119

附录 A

系统过程	125
sp_check_text_index	126
sp_clean_text_events	127
sp_clean_text_indexes	128
sp_create_text_index	129
sp_drop_text_index	131
sp_help_text_index	132
sp_optimize_text_index	133
sp_redo_text_events	134
sp_refresh_text_index	135
sp_show_text_online	136
sp_text_cluster	137
sp_text_configure	139
sp_text_dump_database	140
sp_text_kill	143
sp_text_load_index	144
sp_text_notify	145
sp_text_online	146

附录 B	示例文件	147
	缺省 textsvr.cfg 配置文件	147
	sample_text_main.sql 脚本	151
	演示增强型全文本搜索引擎功能的示例文件	152
	自定义同义词库	152
	主题	152
	集群、汇总和按示例查询	152
	getsend 示例程序	152
附录 C	Unicode 支持	153
附录 D	使用 XML 数据	155
	针对字段和区域正确设置 XML 数据的格式	155
	XML 索引示例	156
	示例片段	156
索引		165

关于本手册

本用户指南描述如何在 Sybase® Adaptive Server® Enterprise 中使用 Enhanced Full-Text Search Specialty Data Store 产品。本书中讲述此产品的增强版的特点和功能。

读者

本书面向那些需要在 Adaptive Server 中配置 Enhanced Full-Text Search Specialty Data Store 的系统管理员，以及那些需要对 Adaptive Server 数据执行全文本搜索的用户。

如何使用本手册

本手册包括以下章节：

- [第 1 章 “简介”](#) 提供对 Enhanced Full-Text Search Specialty Data Store 的概述。
- [第 2 章 “了解增强型全文本搜索引擎”](#) 介绍 Enhanced Full-Text Search Specialty Data Store 的组件及其工作方式。
- [第 3 章 “为全文本搜索配置 Adaptive Server”](#) 介绍如何配置 Adaptive Server 以便 Enhanced Full-Text Search Specialty Data Store 可在数据库上执行全文本搜索。
- [第 4 章 “设置 Verity 功能”](#) 介绍在发布全文本搜索查询之前所需的设置。
- [第 5 章 “编写全文本搜索查询”](#) 介绍用来撰写全文本搜索查询的组件。
- [第 6 章 “系统管理”](#) 提供有关系统管理问题的信息。
- [第 7 章 “性能和调优”](#) 提供有关性能和调优问题的信息。
- [第 8 章 “Verity 主题”](#) 提供有关配置 Verity 引擎的信息。
- [附录 A “系统过程”](#) 介绍 Enhanced Full-Text Search Specialty Data Store 的系统过程。
- [附录 B “示例文件”](#) 提供 *textsvr.cfg* 文件的文本，介绍 Enhanced Full-Text Search Specialty Data Store 附带的示例文件，并讨论有关 *sample_text_main.sql* 脚本的问题。
- [附录 C “Unicode 支持”](#) 介绍如何配置 Enhanced Full Text Search Specialty Data Store 以使用 Unicode。
- [附录 D “使用 XML 数据”](#) 为 XML 数据提供正确的格式，以便其使用字段和区域编制到文本索引中，而且提供使用这种数据的示例。

相关文档

Sybase[®] Adaptive Server[®] Enterprise 文档集由以下文档组成：

- 针对所用平台的发行公告 — 包含未能及时写入手册的最新信息。
最新版本的发行公告可以从万维网上获得。若要查找本产品 CD 发行之后增加的重要产品或文档信息，请使用 Sybase Technical Library。
- 针对所用平台的《安装指南》— 介绍所有 Adaptive Server 产品及相关的 Sybase 产品的安装、升级和配置过程。
- What's New in Adaptive Server Enterprise? (《Adaptive Server Enterprise 中的新增功能》) — 介绍 Adaptive Server 15.0 版的新功能、为支持这些功能而增加的系统更改以及可能会影响现有应用程序的更改。
- ASE Replicator Users Guide (《ASE Replicator 用户指南》) — 介绍如何使用 Adaptive Server 的 Adaptive Server Replicator 功能，实现从主服务器到一个或多个远程 Adaptive Server 的基本复制。
- 《组件集成服务用户指南》— 介绍如何使用 Adaptive Server 组件集成服务功能连接远程 Sybase 数据库和非 Sybase 数据库。
- 针对所用平台的 Configuring Adaptive Server Enterprise (《配置 Adaptive Server Enterprise》) — 提供对 Adaptive Server 执行特定配置任务的操作说明。
- Full-Text Search Specialty Data Store Users Guide (《Full-Text Search Specialty Data Store 用户指南》) — 介绍如何借助 Verity 使用全文本搜索功能对 Adaptive Server Enterprise 数据进行搜索。
- 《词汇表》— 定义 Adaptive Server 文档中使用的技术术语。
- 《Historical Server 用户指南》— 介绍如何使用 Historical Server 来获取 SQL Server[®] 和 Adaptive Server 的性能信息。
- 《Adaptive Server Enterprise 中的 Java》— 介绍如何安装 Java 类以及如何在 Adaptive Server 数据库中将它们用作数据类型、函数和存储过程。
- 《Job Scheduler 用户指南》— 提供有关如何使用命令行或图形用户界面 (GUI) 在本地或远程 Adaptive Server 上进行安装和配置以及创建和调度作业的操作说明。
- 《Monitor Client Library 程序员指南》— 介绍如何编写可访问 Adaptive Server 性能数据的 Monitor Client Library 应用程序。
- 《Monitor Server 用户指南》— 介绍如何使用 Monitor Server 从 SQL Server 和 Adaptive Server 获取性能统计信息。

- **Performance and Tuning Guide**（《性能和调优指南》）— 是一套共计 4 本的系列丛书，介绍如何调整 Adaptive Server 以获得最优性能：
 - **Basics**（《基础知识》）— 介绍了解和研究 Adaptive Server 的性能问题需具备的基础知识。
 - **Locking**（《锁定》）— 介绍如何使用各种锁定模式提高 Adaptive Server 的性能。
 - **Optimizer and Abstract Plans**（《优化程序和抽象计划》）— 介绍优化程序如何处理查询以及如何使用抽象计划更改某些优化程序计划。
 - **Monitoring and Analyzing**（《监控和分析》）— 介绍如何获得统计信息以及如何使用统计信息监控和优化性能。
- 《快速参考指南》— 这是一本袖珍手册，完整地列出了各种命令、函数、系统过程、扩展系统过程、数据类型和实用程序的名称和语法。
- 《参考手册》— 是一套共计 4 本的系列丛书，详细介绍了下列 Transact-SQL[®] 信息：
 - 《构件块》— Transact-SQL 数据类型、函数、全局变量、表达式、标识符、通配符和保留字。
 - 《命令》— 各种 Transact-SQL 命令。
 - 《过程》— Transact-SQL 系统过程、目录存储过程、系统扩展存储过程和 dbcc 存储过程。
 - 《表》— Transact-SQL 系统表和 dbcc 表。
- 《系统管理指南》— 提供有关管理服务器和数据库的详细信息。本手册包括针对管理物理资源、安全性、用户数据库和系统数据库以及指定字符转换、国际语言和排序顺序设置的说明和指南。
- **System Tables Diagram**（《系统表框图》）— 以张贴画的形式阐明系统表及其实体关系。只提供印刷手册。
- 《Transact-SQL 用户指南》— 提供有关 Transact-SQL 这一 Sybase 的关系数据库语言增强版的文档资料。本手册可用作数据库管理系统初级用户的教科书。本手册还包括对 pubs2 和 pubs3 样本数据库的说明。
- **Using Adaptive Server Distributed Transaction Management Features**（《使用 Adaptive Server 分布式事务管理功能》）— 介绍如何在分布式事务处理环境中配置、使用 Adaptive Server DTM 功能以及如何排除其中的故障。

-
- 《在高可用性系统中使用 Sybase 故障切换》— 提供有关使用 Sybase 的故障切换功能将 Adaptive Server 配置为高可用性系统中的协同服务器的说明。
 - Unified Agent and Agent Management Console (《Unified Agent 和 Agent Management Console》) — 介绍提供运行时服务以管理、监控和控制分布式 Sybase 资源的 Unified Agent。
 - 《实用程序指南》— 提供有关在操作系统级别执行的 Adaptive Server 实用程序 (如 isql 和 bcp) 的文档资料。
 - 《Web 服务用户指南》— 介绍如何配置、使用 Adaptive Server Web 服务以及如何排除其中的故障。
 - 《适用于 CICS、Encina 和 TUXEDO 的 XA 接口集成指南》— 提供有关将 Sybase DTM XA 接口与 X/Open XA 事务管理器配合使用的说明。
 - 《Adaptive Server Enterprise 中的 XML 服务》— 介绍 Sybase 本机 XML 处理器和 Sybase 基于 Java 的 XML 支持以及数据库中的 XML, 并提供有关构成 XML 服务的查询和映射函数的文档资料。

其它信息来源

使用 Sybase Getting Started CD、SyBooks CD 和 Sybase Product Manuals 网站可以了解有关产品的更多信息:

- Getting Started CD 包含 PDF 格式的发行公告和安装指南, 也可能包含 SyBooks CD 中未收纳的其它文档或更新信息。它随软件一起提供。若要阅读或打印 Getting Started CD 上的文档, 需要使用 Adobe Acrobat Reader, 该软件可以通过 CD 上提供的链接从 Adobe 网站免费下载。
- SyBooks CD 含有产品手册, 随软件提供。通过使用基于 Eclipse 的 SyBooks 浏览器, 您可以阅读简单易用且基于 HTML 的格式的手册。有些文档可能是以 PDF 格式提供的, 您可以通过 SyBooks CD 上的 PDF 目录访问这些文档。若要阅读或打印 PDF 文件, 需要使用 Adobe Acrobat Reader。

有关安装和启动 SyBooks 的操作说明, 请参考 Getting Started CD 上的《SyBooks 安装指南》或 SyBooks CD 上的 *README.txt* 文件。

- Sybase Product Manuals 网站是 SyBooks CD 的联机版本，您可以使用标准 Web 浏览器进行访问。除了产品手册之外，还可以找到有关“EBF/ 维护” (EBFs/Maintenance)、 “技术文档” (Technical Documents)、 “案例管理” (Case Management)、 “已解决的案例” (Solved Cases)、 “新闻组” (Newsgroups) 和 “Sybase 开发人员网络” (Sybase Developer Network) 的链接。

若要访问 Sybase Product Manuals Web 站点，请转到位于 <http://www.sybase.com/support/manuals/> 的“产品手册” (Product Manuals)。

Web 上的 Sybase 认证 Sybase Web 站点上的技术文档不断在更新。

❖ **查找有关产品认证的最新信息**

- 1 将 Web 浏览器定位到位于 <http://www.sybase.com/support/techdocs/> 的“技术文档” (Technical Documents)。
- 2 从左边的导航栏中选择“产品” (Products)。
- 3 从产品列表中选择一种产品名称并单击“查找” (Go)。
- 4 选择“认证报告” (Certification Report) 过滤器，指定时间范围，然后单击“查找” (Go)。
- 5 单击“认证报告” (Certification Report) 标题显示此报告。

❖ **查找有关组件认证的最新信息**

- 1 将 Web 浏览器定位到位于 <http://certification.sybase.com/> 的“可用性和认证报告” (Availability and Certification Reports)。
- 2 在“按产品搜索” (Search by Products) 下选择产品系列和产品；或在“按平台搜索” (Search by Platform) 下选择平台和产品。
- 3 选择“搜索” (Search) 以显示所选项目的可用性和认证报告。

❖ **创建 Sybase Web 站点（包括支持页）的个人化视图**

建立 MySybase 配置文件。MySybase 是一项免费服务，它允许您创建 Sybase Web 页的个人化视图。

- 1 将 Web 浏览器定位到位于 <http://www.sybase.com/support/techdocs/> 的“技术文档” (Technical Documents)。
- 2 单击“我的 Sybase” (MySybase) 并创建 MySybase 配置文件。

Sybase EBF 和软件维护

❖ 查找有关 EBF 和软件维护的最新信息

- 1 将 Web 浏览器定位到位于 <http://www.sybase.com/support> 的“Sybase 支持页” (Sybase Support Page)。
- 2 选择“EBF/ 维护” (EBFs/Maintenance)。如果出现提示信息，请输入您的 MySybase 用户名和口令。
- 3 选择一个产品。
- 4 指定时间范围并单击“查找” (Go)。将显示 EBF/ 维护的版本列表。
锁形图标表示由于您没有注册为“技术支持联系人” (Technical Support Contact)，因此您没有某些 EBF/ 维护版本的下载授权。如果您尚未注册，但拥有 Sybase 代表提供的或通过支持合同获得的有效信息，请单击“编辑角色” (Edit Roles) 将“技术支持联系人” (Technical Support Contact) 角色添加到 MySybase 配置文件中。
- 5 单击信息图标可显示 EBF/ 维护报告，单击产品说明可下载该软件。

约定

目录路径 为提高可读性，本手册的目录路径采用 UNIX 格式。在 Windows NT 中，请用 %SYBASE% 替换 \$SYBASE，并用反斜杠 (\) 替换斜杠 (/)。例如，将以下用户输入：

```
$SYBASE/$SYBASE_FTS/scripts
```

替换为：

```
%SYBASE%\%SYBASE_FTS%\scripts
```

格式化 SQL 语句 SQL 是一种自由格式语言：没有对每行中可容纳的单词数量以及应在何处折行作出规定。不过，为了便于阅读，本手册中的所有示例和语法语句都采用统一的格式，即一个语句的每个子句都启用一个新行。有多个成分的子句会扩展到其它行，这些行会有缩进。

SQL 语法定义 本手册中语法语句的约定如下：

表 1: 语法语句约定

凡例	定义
command	命令名、命令选项名、实用程序名、实用程序标记和其它关键字在语法语句中用 bold Courier 字体显示；在段落文本中用 bold Helvetica 字体显示。
variable	变量（即代表要填充的值的词语）用 <i>斜体</i> 显示。
{ }	大括号表示至少选择括号中的一个选项。不要在选项中包括大括号。
[]	中括号表示可以选择括号中的一个或多个选项，也可以不选。不要在选项中包括中括号。
()	小括号可作为命令的一部分输入。
	竖线表示只可以选择一个显示的选项。
,	逗号表示可以选择任意多个显示的选项，可输入逗号作为命令的一部分来分隔选项。

- 语法语句（显示命令的语法和所有选项）显示如下：

```
sp_dropdevice [device_name]
```

或者，对于具有多个选项的命令：

```
select column_name
from table_name
where search_conditions
```

在语法语句中，关键字（命令）采用常规字体，而标识符使用小写：关键字采用常规字体，用户提供的内容用斜体表示。

- 说明 Transact-SQL 命令用法的示例如下：

```
select * from publishers
```

- 计算机的输出示例显示如下：

pub_id	pub_name	city	state
-----	-----	-----	-----
0736	New Age Books	Boston	MA
0877	Binnet & Hardley	Washington	DC
1389	Algodata Infosystems	Berkeley	CA

(3 rows affected)

大小写 本手册中，大多数示例是小写字体。不过，输入 Transact-SQL 关键字时可以忽略大小写。例如，**SELECT**、**Select** 和 **select** 是相同的。

Adaptive Server 是否区分数据库对象（如表名）的大小写，取决于安装在 Adaptive Server 上的排序顺序。通过重新配置 Adaptive Server 的排序顺序，可以更改单字节字符集的区别大小写设置。

必选选项 { 必须至少选择一个 } • *大括号和竖线*: 仅选择一个选项:

```
{die_on_your_feet | live_on_your_knees | live_on_your_feet}
```

- *大括号和逗号*: 选择一个或多个选项。如果选择多个选项，请使用逗号分隔所选选项。

```
{cash, check, credit}
```

可选项 [并非必须选择] • *中括号中的一项*: 不必选择此项。

```
[anchovies]
```

- *中括号和竖线*: 一个都不选或仅选择一个。

```
[beans | rice | sweet_potatoes]
```

- *中括号和逗号*: 选择零个、一个或多个选项。如果选择多个选项，请使用逗号分隔所选选项。

```
[extra_cheese, avocados, sour_cream]
```

省略号: 再次使用 (反复使用) ... 省略号 (...) 表示您可以根据需要任意重复最后一个单元。在以下语法语句中，**buy** 是必需的关键字:

```
buy thing = price [cash | check | credit]
[, thing = price [cash | check | credit]]...
```

您必须至少购买一件产品，并给出其价格。您可以选择方括号中列出的某种付款方式。您还可以选购其它产品: 购买数量随意。对于要买的每种产品，给出其名称、价格和付款方式 (可选)。

辅助功能特性

本文档具有为提供辅助功能而进行了专门设计的 HTML 版本。可以利用相应的技术 (如屏幕阅读器) 浏览 HTML，也可以用屏幕放大器进行查看。

此版本的 Enhanced Specialty Data Store 和 HTML 文档已经过测试，符合美国政府“第 508 节辅助功能”的要求。符合“第 508 节”要求的文档一般也符合非美国的辅助功能原则，如 World Wide Web 协会 (W3C) 针对 Web 站点的原则。

此产品的联机帮助也是以 HTML 格式提供的，您可以使用屏幕阅读器进行浏览。

注释 您可能需要对辅助功能工具进行配置以实现最优化。某些屏幕阅读器按照大小写来辨别文本，例如将“ALL UPPERCASE TEXT”看作首字母缩写，而将“MixedCase Text”看作单词。对工具进行配置，规定语法约定，您可能会感觉更方便。有关工具的信息，请查阅文档。

有关 Sybase 如何支持辅助功能的信息，请参见位于 <http://www.sybase.com/accessibility> 的“Sybase 辅助功能” (Sybase Accessibility)。Sybase 辅助功能站点包括指向“第 508 节”和 W3C 标准相关信息的链接。



简介

Enhanced Full-Text Search Specialty Data Store（本手册中称为“增强型全文本搜索引擎”）是一个 Open Server™ 应用程序，它是基于 Verity 开发人员工具包中提供的 Verity 技术开发的。Adaptive Server 通过组件集成服务 (CIS) 与增强型全文本搜索引擎相连接，从而允许以 Verity 查询语言编写的查询对 Adaptive Server 数据执行全文本搜索。

本手册描述 Enhanced Full-Text Search Specialty Data Store 的特点和功能。

主题	页码
增强型全文本搜索引擎的功能	1
高可用性	2

增强型全文本搜索引擎的功能

Enhanced Full-Text Search Specialty Data Store 产品可以对 Adaptive Server 数据执行强大的全文本搜索。在 Adaptive Server 中，如果没有增强型全文本搜索引擎，那么，在搜索文本列时，只能搜索与 `select` 语句所指定的条件相匹配的数据。例如，如果某个表包含有关犬种的文档，在执行对单词“Saint Bernard”的搜索时，该查询仅可得到文本列中包含“Saint Bernard”的行。

有了增强型全文本搜索引擎，则可扩展对文本列的查询以便执行以下操作：

- 按搜索项在所选文档中出现的频率高低排列结果。例如，可以获得文档中出现单词“Saint Bernard”达到五次或更多次的文档标题的列表。
- 选择要搜索的单词相距不到 n 个单词距离的文档。例如，可搜索包含单词“Saint Bernard”和“Swiss Alps”且这两个单词出现在连续的 10 个单词之内的文档。

- 选择在一个段落或一个句子之中包含了指定的所有搜索元素的文档。例如，可以查询在相同的段落或句子中包含单词 “Saint Bernard” 和 “Swiss Alps” 的文档。
- 选择包含所指定单词的一个或多个同义词的文档。例如，选择讨论 “dogs” 的文档时会返回包含 “dogs”、“canine”、“pooch”、“pup” 等单词的文档。
- 创建自定义同义词库。例如，可创建一个自定义同义词库，在其中包括 “working dogs”、“St.Bernard”、“large dogs” 和 “European Breeds” 作为 “Saint Bernard” 的同义词。
- 创建可指定查询的搜索条件的主题。例如，可以创建一个主题，用于返回包含短语 “Saint Bernard” 或 “St.Bernard” 的文档，并接着返回包含短语 “working dogs”、“large dogs” 或 “European Breeds” 的文档。
- 返回按集群分组的文档，以便您了解文档所讨论的主要主题。
- 在文档中选择相关文本的某一部分，并搜索其它相似文档。
- 生成多种不同文档类型的索引，如 Microsoft Word 和 FrameMaker。
- 使用多达 16 种排序顺序对文档进行排序。
- 集成本地和恢复功能。
- 使用系统过程更改配置参数的值。
- 在服务器处于非活动状态时，优化文本搜索索引，以提高性能。
- 创建查看设置信息的其它系统管理报告。
- 能够使数据库自动联机以进行文本搜索。

高可用性

增强型全文本搜索产品支持 Sybase 故障切换。如果 Adaptive Server 出现故障，增强型全文本搜索将接受来自协同服务器的连接。另外，如果 Adaptive Server 启用了代理数据库支持，那么主服务器和协同服务器可同时使用增强型全文本搜索。

了解增强型全文本搜索引擎

本章介绍增强型全文本搜索引擎的工作方式。

主题	页码
增强型全文本搜索引擎的组件	3
全文本搜索的工作方式	8

增强型全文本搜索引擎的组件

增强型全文本搜索引擎使用以下组件提供全文本搜索功能：

- 源表
- Verity 集合（文本索引）
- 不同文档类型的过滤器
- text_db 数据库
- 索引表
- text_events 表

源表

源表是由 Adaptive Server 维护的用户表。它包含一个或多个 date、time、text、image、char、varchar、datetime、small datetime、bigint、int、smallint、tinyint、unsigned bigint、unsigned int、unsigned smallint 或 unitext 数据类型的列，存放要搜索的数据。源表必须有一个 IDENTITY 列或主键，用来在文本搜索过程中将源表与索引表的 id 列连接。

源表可以是存放实际数据的本地表，也可以是使用 CIS 映射到远程数据的代理表。

Verity 集合

增强型全文本搜索引擎使用位于 `$$SYBASE/$$SYBASE_FTS/collections` 中的 Verity 集合。当您按照第 18 页的“创建文本索引和索引表”中的介绍创建文本索引时，Verity 会创建一个集合，这是执行文本索引的目录。增强型全文本搜索引擎将查询该集合。有关 Verity 集合的详细信息，请访问位于 <http://www.verity.com> 的 Verity 网站。

过滤器

文本索引使用过滤器来剔除文档中非 ASCII 文本的标记。增强型全文本搜索引擎为各种文档类型（Microsoft Word、PDF、WordPerfect、SGML 和 HTML）提供了过滤器。

text_db 数据库

在安装增强型全文本搜索引擎的过程中，可使用安装脚本 `installtextserver` 将名为 `text_db` 的数据库添加到 Adaptive Server 中，如第 12 页的“运行 `installtextserver` 脚本”中所述。该数据库不包含任何用户数据，但包含两个支持表：`vesaux` 和 `vesauxcol`。这些表包含增强型全文本搜索引擎使用的元数据，以保持 Adaptive Server 源表和 Verity 集合的完整性。

如果在对索引列执行了 `insert`、`update` 或 `delete` 命令后更新集合，增强型全文本搜索引擎会查询 `vesaux` 和 `vesauxcol` 表。这些表确定哪些集合包含修改后的列，以便所有受影响的集合都得到更新。增强型全文本搜索引擎在联机时也会使用这些表，以确保所有必需的集合都存在。

vesaux 表

`vesaux` 表中的列在表 2-1 中介绍。

表 2-1: vesaux 表中的列

列名	说明
<code>id</code>	IDENTITY 列
<code>object_name</code>	创建外部索引的源表的名称
<code>option_string</code>	文本索引创建选项
<code>collection_id</code>	Verity 集合的名称
<code>key_column</code>	源表中的 IDENTITY 列或主键的名称
<code>svrid</code>	维护集合的增强型全文本搜索引擎的服务器 ID

vesauxcol 表

vesauxcol 表中的列在表 2-2 中介绍。

表 2-2: vesauxcol 表中的列

列名	说明
id	vesaux 表中被引用行的 ID
col_name	要搜索的列的名称
col_type	列类型 (date、time、text、image、char、varchar、datetime 和 smalldatetime；对于增强型全文本搜索引擎，还包括 int、smallint 和 tinyint)

index 表

索引表提供了一种定位和搜索源表中存储的文档的方式。index 表由增强型全文本搜索引擎维护，并有一个 id 列映射到对应的源表的 IDENTITY 列或主键。源表中行的 IDENTITY 值或主键值随同 Verity 集合中的数据一起存储，这样就可以连接源表和索引表。尽管索引表由增强型全文本搜索引擎存储和维护，但它做为代理表通过组件集成服务可链接 Adaptive Server。

索引表包含有一些名为 *伪列* 的特殊列。增强型全文本搜索引擎使用这些特殊列来确定搜索的参数以及文本数据在源表中的位置。伪列没有关联的物理存储区，伪列的值只在查询期间有效，并在运行完查询后立即被删除。

例如，当在查询中使用 *score* 伪列，按照文档与查询的匹配程度对每个文档进行分级时，可能需要使用一个值为 15 的 *score*，在文本数据库中查找短语 “small Saint Bernards”。这个短语不常出现，因此低的 *score* 值可扩展搜索范围以包括零星出现满足搜索条件的项的文档。但是，如果搜索的是一个常见的短语，例如 “large Saint Bernards”，则可以使用值为 90 的 *score*，这样将搜索限制为那些大量出现搜索条件的文档。

可使用 *score* 列和其它一些伪列，如 *id*、*index_any*、*sort_by*、*summary* 和 *max_docs*，指定要在搜索中包括的参数。有关于伪列的说明，请参见第 42 页的“索引表中的伪列”。

text_events 表

对于每个包含存在文本索引的表的数据库而言，都必须包含一个**事件表**，记录索引列的插入、更新和删除。该表的名称是 `text_events`，该表用于将更新数据传播到 Verity 集合。

`text_events` 表中的列在表 2-3 中介绍。

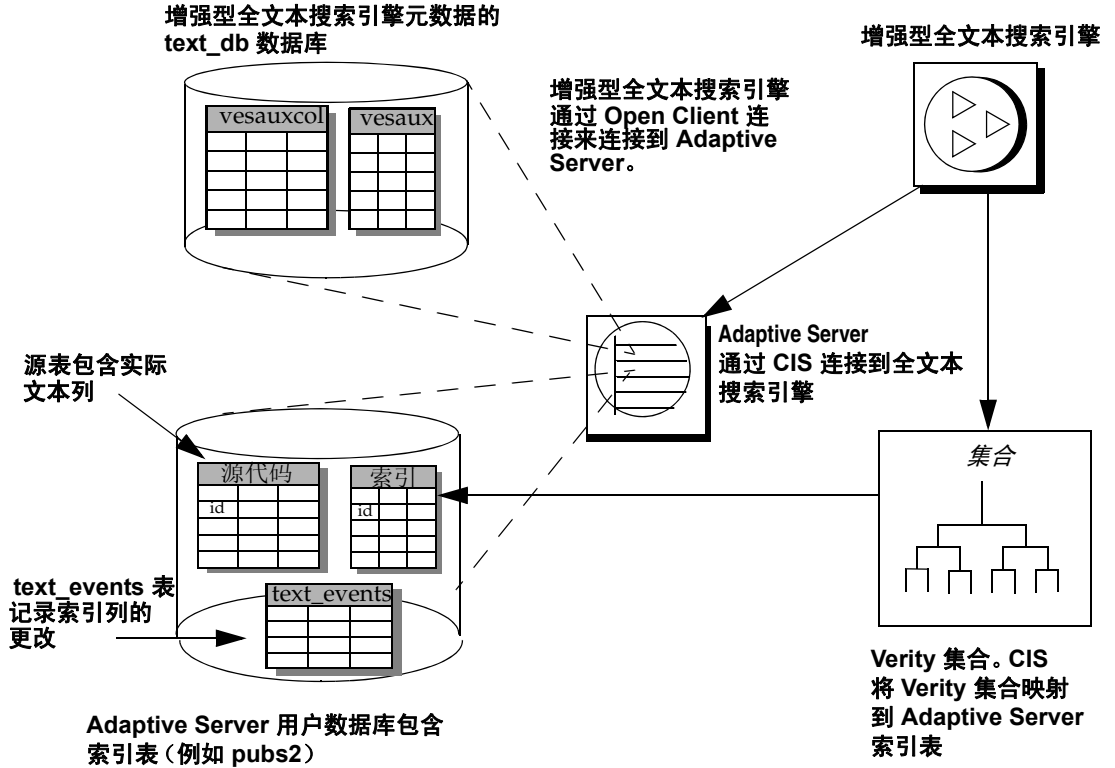
表 2-3: text_events 表中的列

列名	说明
event_id	IDENTITY 列
id	更新、插入或删除的行的 ID
tableid	包含更新、插入或删除的行的表的名称
columnid	创建 text 索引的列的名称
event_date	执行 update、insert 或 delete 操作的日期和时间
event_type	更新类型（update、insert 或 delete）
event_status	指示是否已将 update、insert 或 delete 的结果传播到集合中 <ul style="list-style-type: none">• 0 — 未读事件• 1 — 已读事件• 2 — 事件成功• 3 — 事件失败
srvid	维护集合的增强型全文搜索引擎的服务器 ID

组件之间的关系

增强型全文搜索引擎组件之间的关系显示在图 2-1 中。

图 2-1: 增强型全文搜索引擎的组件



全文本搜索的工作方式

若要执行全文本搜索，应输入一条 `select` 语句，将源表的 `IDENTITY` 列或主键与索引表的 `id` 列连接在一起，并根据需要使用伪列来定义搜索。例如，以下查询在 `pubs2` 数据库的 `blurbs` 表中搜索单词 “Greek” 出现在单词 “Gustibus” 附近的文档（`i_blurbs` 表是索引表）：

```
select t1.score, t2.copy
from i_blurbs t1, blurbs t2
where t1.id=t2.id and t1.score > 20
and t1.max_docs = 10
and t1.index_any = "<near>(Greek, Gustibus)"
```

Adaptive Server 和增强型全文本搜索引擎将查询过程分解为如下具体步骤：

- 1 增强型全文本搜索引擎处理查询：

```
select t1.score, t1.id
from i_blurbs t1
where t1.score > 20
and t1.max_docs = 10
and t1.index_any = "<near>(Greek, Gustibus)"
```

`select` 语句包含 Verity 运算符 `near` 以及伪列 `score`、`max_docs` 和 `index_any`。运算符和伪列为在 Verity 集合上进行的搜索提供参数，它们将结果集从整个 `copy` 列限制为文档中单词 “Greek” 和 “Gustibus” 彼此最接近的 10 个文档。

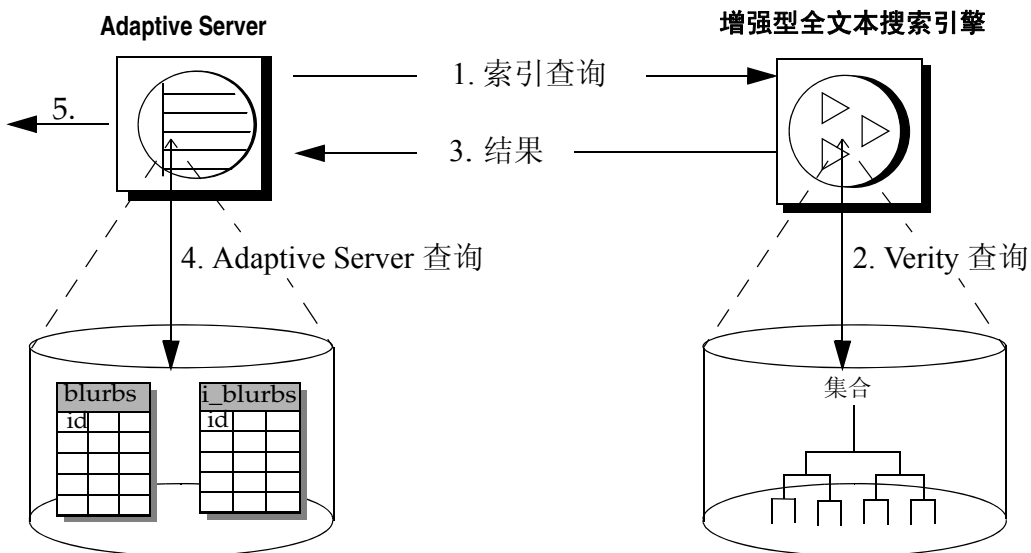
- 2 Adaptive Server 对步骤 1 中由增强型全文本搜索引擎返回的结果集执行下面的 `select` 语句：

```
select t1.score, t2.copy
from i_blurbs t1, blurbs t2
where t1.id=t2.id
```

这样就通过 `blurbs` 表的 `IDENTITY` 列或主键和 `i_blurbs` 表的 `id` 列，将 `blurbs` 表（源表）和 `i_blurbs` 表（索引表）连接在一起。

图 2-2 描述了 Adaptive Server 和增强型全文本搜索引擎处理查询的方式。

图 2-2: 处理全文本搜索查询



本章介绍如何配置 Adaptive Server 执行全文本搜索。

主题	页码
为增强型全文本搜索引擎配置 Adaptive Server	11
创建和维护文本索引	16

为增强型全文本搜索引擎配置 Adaptive Server

增强型全文本搜索引擎是一个远程服务器，Adaptive Server 通过组件集成服务 (CIS) 与之连接。在使用增强型全文本搜索引擎之前，必须为它配置 Adaptive Server，方法如下：

- 启用 `enable cis`、`cis rpc handling` 和 `full-text search` 配置参数（如果尚未启用）。启用全文本搜索时需要许可证。
- 运行 `installtextserver` 脚本，定义一个或多个增强型全文本搜索引擎。
- 运行 `installmessages` 脚本，为增强型全文本搜索引擎的系统过程安装消息。
- 运行 `installevent` 脚本，在每个将包含文本索引的用户数据库中创建 `text_events` 表。
- 命名本地服务器并重新启动。

启用配置参数

若要连接到增强型全文本搜索引擎，必须在启用 `enable cis` 和 `cis rpc handling` 配置参数的情况下运行 Adaptive Server。如果没有启用这些参数，请使用 `isql` 登录到 Adaptive Server，然后用 `sp_configure` 启用它们。例如：

```
exec sp_configure "enable cis", 1
exec sp_configure "cis rpc handling", 1
exec sp_configure "enable full-text search", 1
```

如果对 `enable cis` 进行了更改，则会显示消息，指出必须重新启动 Adaptive Server 才能使新配置参数生效。

运行 `installtextserver` 脚本

`installtextserver` 脚本将执行以下操作：

- 将增强型全文本搜索引擎定义为一个连接到 Adaptive Server 的远程服务器，其服务器类为 `sds`。
- 创建存储文本索引元数据的数据库。有关此数据库的详细信息，请参见第 4 页的“[text_db 数据库](#)”。
- 安装增强型全文本搜索引擎所需的系统过程。

只运行一次 `installtextserver` 脚本（请参见第 14 页的“[启动 installtextserver 脚本](#)”）。以后要添加增强型全文本搜索引擎，请使用 `sp_addserver`。有关 `sp_addserver` 的详细信息，请参见第 77 页的“[配置多个增强型全文本搜索引擎](#)”。

注释 如果要将 `installtextserver` 脚本放到 `model` 数据库上，则在运行该脚本之前，必须将 `model` 数据库的大小增加到至少 3MB。

所有增强型全文本搜索引擎都使用同一个数据库来存储文本索引元数据。该数据库也称作 `text_db`，这是它的缺省名称。

有关使用 `installtextserver` 脚本添加系统过程的列表和说明，请参见附录 A “[系统过程](#)”

编辑 `installtextserver` 脚本

`installtextserver` 脚本位于 `$$SYBASE/$SYBASE_FTS/scripts` 目录中。使用文本编辑器（例如 `vi` 或 `emacs`）可打开并编辑该脚本。可进行如下编辑：

- 更改 `text_db` 数据库的名称。如果使用了不同的名称，请用相应的名称替换所有的 `text_db` 名称。

注释 如果更改 `text_db` 数据库的名称，则必须同时更改 `defaultDb` 配置参数中的名称（请参见第 62 页的“修改配置参数”）。

- 更改增强型全文本搜索引擎的名称。缺省情况下，`installtextserver` 脚本定义名为“`textsvr`”的增强型全文本搜索引擎。如果对增强型全文本搜索引擎使用了其它名称，请编辑该脚本，以便它定义正确的服务器名称。
- 添加多个增强型全文本搜索引擎（有关这样做如何提高性能的信息，请参见第 77 页的“配置多个增强型全文本搜索引擎”）。如果最初定义了多个增强型全文本搜索引擎，则请编辑 `installtextserver` 脚本，使其包含所有增强型全文本搜索引擎的定义。`installtextserver` 中包含以下部分以命名您正在配置的增强型全文本搜索引擎（缺省为“`textsvr`”）：

```
/*
** Add the text server
*/
exec sp_addserver textsvr,sds,textsvr
go
```

为要配置的每个增强型全文本搜索引擎添加一个条目。例如，如果要配置三个名称分别为 `KRAZYKAT`、`OFFICAPUP` 和 `MOUSE` 的增强型全文本搜索引擎，请用以下行替换缺省的“`textsvr`”行：

```
exec sp_addserver KRAZYKAT, sds, KRAZYKAT
exec sp_addserver OFFICAPUP, sds, OFFICAPUP
exec sp_addserver MOUSE, sds, MOUSE
go
```

- 如果使用 `OmniConnect` 与增强型全文本搜索引擎通信，请在 `sp_addobjectdef` 对 `vesaux` 和 `vesauxcol` 表的调用中将服务器名称更改为有效的远程服务器。例如，如果远程服务器名为 `REMOTE`，应将以下行：

```
exec sp_addobjectdef
"vesaux", "SYBASE.master.dbo.vesaux", "table"
exec sp_addobjectdef
"vesauxcol", "SYBASE.master.dbo.vesauxcol",
```

```
"table"
```

更改为:

```
exec sp_addobjectdef
"vesaux", "REMOTE.master.dbo.vesaux", "table"
exec sp_addobjectdef
"vesauxcol", "REMOTE.master.dbo.vesauxcol",
"table"
```

启动 *installtextserver* 脚本

使用 `isql` 运行 `installtextserver` 脚本。例如, 若要在名为 `MYSVR` 的 Adaptive Server 中运行 `installtextserver` 脚本, 请输入:

```
isql -Usa -P -SMYSVR -i
$SYBASE/$SYBASE_FTS/scripts/installtextserver
```

运行 *installmessages* 脚本

增强型全文搜索引擎有自己的系统过程消息集, 必须在 Adaptive Server 中安装这些消息。可使用 `installmessages` 脚本安装这些消息。即使有多个增强型全文搜索引擎, 也只需运行一次 `installmessages` 脚本。

例如, 要在名为 `MYSVR` 的服务器中运行 `installmessages` 脚本, 请输入:

```
isql -Usa -P -SMYSVR -i
$SYBASE/$SYBASE_FTS/scripts/installmessages
```

运行 *installevent* 脚本

每个包含由文本索引引用的表的数据库都必须包含一个 `text_events` 表, 以记录索引列的插入、更新和删除。该表用于将更新数据传播到 Verity 集合。

按如下所述运行 `installevent` 脚本, 可在数据库中创建 `text_events` 表和关联的系统过程。按如下所述使用 `installevent` 脚本:

- 如果所有的数据库都需要文本索引, 请运行 `installevent` 脚本以便在 `model` 数据库中创建一个 `text_events` 表。这样, 每个新建的数据库都将有一个 `text_events` 表。若要向现有的数据库添加一个 `text_events` 表, 请按如下所述编辑脚本以便在现有的用户数据库中创建 `text_events` 表。

- 如果并非所有的数据库都需要文本索引，则可使用 `installevent` 脚本作为一个样本。对每个包含需要文本索引的表的现有数据库和新数据库，运行 `installevent` 脚本。必须按如下所述编辑脚本，以便在正确的用户数据库中创建 `text_events` 表。

注释 如果在包含需要文本索引的源表的数据库中不存在 `text_events` 表，则对源表的更改将不会传播到 Verity 集合。

编辑 `installevent` 脚本

`installevent` 脚本位于 `$$SYBASE/$SYBASE_FTS/scripts` 目录中。使用文本编辑器（例如 `vi` 或 `emacs`）打开脚本并进行编辑。可进行如下编辑：

- 更改用户数据库的名称。`installevent` 脚本在 `model` 数据库中创建事件表（名为 `text_events`）和相关联的系统过程。`model` 数据库是缺省数据库。若要在现有的用户数据库中安装 `text_events` 表，请编辑脚本，用用户数据库名替换所有的 `model`。
- 更改 `text_db` 数据库的名称。如果存储文本索引元数据的数据库不是用 `text_db` 命名的，则用适当的名称替换所有对 `text_db` 的引用。

注释 `text_db` 数据库的名称必须与在 `defaultDb` 配置参数中的名称相同（请参见第 62 页的“修改配置参数”）。

运行 `installevent` 脚本

使用 `isql` 运行 `installevent` 脚本可在 Adaptive Server 中安装 `text_events` 表和相关的系统过程。例如，若要在名为 `MYSVR` 的服务器中运行 `installevent` 脚本，请输入：

```
isql -Usa -P -SMYSVR -i
$$SYBASE/$SYBASE_FTS/scripts/installevent
```

注释 `text_db` 数据库必须在运行 `installevent` 脚本之前就已存在。如果不存在，则应先运行 `installtextserver` 脚本。

注释 在 64 位平台上安装 `EFTS-12_5` 脚本（`installtextserver`、`installevents`、`installmessages`）之前，应将 `tempdb` 和 `model` 数据库的大小从缺省值增加到 3MB。

命名本地 Adaptive Server

将增强型全文本搜索引擎与 Adaptive Server 12.5 及更高版本一起使用时，必须使用 `sp_addserver <servername>, local` 命名本地 Adaptive Server。发出 `sp_addserver` 之后，必须重新启动本地 Adaptive Server。不要在 `model` 数据库中安装任何系统存储过程，而应安装在 `sybsystemprocs` 中。

创建和维护文本索引

必须为用户数据库中的源表创建文本索引，增强型全文本搜索引擎才能处理全文本搜索。创建文本索引后，当源数据更改时，必须更新文本索引以使文本索引保持最新。若要创建和维护文本索引，请执行以下操作：

- 1 设置索引源表（请参见第 17 页的“为源表设置索引”）。
- 2 创建文本索引和索引表（请参见第 18 页的“创建文本索引和索引表”）。
- 3 使数据库联机以进行全文本搜索（请参见第 20 页的“使数据库联机以进行全文本搜索”）。
- 4 将用户数据中的更改传播到文本索引（请参见第 20 页的“将更改传播到文本索引”）。
- 5 如果要复制文本索引，请设置目标数据库的文本索引（请参见第 21 页的“复制文本索引”）。

有关设置文本索引的示例，请参见 `$$SYBASE/$$SYBASE_FTS/sample/scripts` 目录中的示例脚本 `sample_text_main.sql`。

为源表设置索引

源表包含搜索所基于的数据（例如，pubs2 数据库中的 blurbs 表）。有关详细信息，请参见第 3 页的“源表”。

在源表上创建文本索引前，必须：

- 确认源表具有 IDENTITY 列或主键。如果没有，请更改该表以添加 IDENTITY 列。
- 在主键或 IDENTITY 列上创建唯一索引（可选）。

每个源表都必须包含 IDENTITY 列或主键，以唯一标识每一行并提供连接索引表和源表的方法。创建文本索引时，IDENTITY 列或主键与索引列一起传递给增强型全文本搜索引擎。IDENTITY 列或主键值存储在文本索引中，并映射到索引表中的 id 列。如果表没有主键，可以向该表添加 IDENTITY 列。

向源表添加 IDENTITY 列

若要在名为 composers 的表中创建 IDENTITY 列，请按如下所示定义该表：

```
create table composers (
    id          numeric(m,n)      identity,
    comp_fname  char(30)          not null,
    comp_lname  char(30)          not null,
    text_col    text
)
```

其中， $m \leq 38$ 且 n 始终为 0

若要向现有表中添加 IDENTITY 列，请输入：

```
alter table table_name add id numeric(10,0) identity
```

向 IDENTITY 列中添加唯一索引

为获得最佳性能，Sybase 建议在 IDENTITY 列上创建唯一索引。例如，若要在上面所创建的 IDENTITY 列上创建一个名为 comp_id 的唯一索引，请输入：

```
create unique index comp_id
on composers(id)
```

有关创建唯一索引的详细信息，请参见《Transact-SQL 用户指南》的第 11 章“创建表的索引”。

创建文本索引和索引表

使用 `sp_create_text_index` 可创建文本索引。`sp_create_text_index` 将执行以下操作：

- 更新 `text_db` 数据库中的 `vesaux` 和 `vesauxcol` 表。
- 创建文本索引（Verity 集合）
- 填充 Verity 集合
- 在源表所在的用户数据库中创建索引表

注释 若要成功执行 `sp_create_text_index`，必须运行增强型全文本搜索引擎。有关启动和停止增强型全文本搜索引擎的信息，请参见第 6 章“系统管理”。

文本索引最多可包含 16 列。可以对以下数据类型的列编制索引：`char`、`varchar`、`nchar`、`nvarchar`、`date`、`time`、`text`、`image`、`datetime`、`smalldatetime`、`int`、`smallint`、`tinyint`、`unichar` 和 `univarchar`。

例如，若要在 `KRAZYKAT` 上为 `pubs2` 中 `blurbs` 表的 `copy` 列创建文本索引和一个名为 `i_blurbs` 的索引表，请输入：

```
sp_create_text_index "KRAZYKAT", "i_blurbs", "blurbs", " ", "copy"
```

其中：

- `KRAZYKAT` 是增强型全文本搜索引擎的名称。
- `i_blurbs` 是要创建的索引表和文本索引的名称。
- `blurbs` 是创建文本索引的源表。
- `" "` 是文本索引创建选项的占位符。
- `copy` 是要索引的 `blurbs` 表中的列。

有关详细信息，请参见第 129 页的“`sp_create_text_index`”。

注释 确保配置文件（列在 `defaultDb` 参数之后）中的 `text_db` 数据库名与 Adaptive Server 中的数据库名匹配。如果不匹配，则不能创建文本索引。另外，确保用户数据库中存在 `text_events` 表。如果此表不存在，请为该数据库运行 `installevent` 脚本（请参见第 14 页的“运行 `installevent` 脚本”）。

传播 Verity 集合可能只需要几分钟，也可能需要数小时，具体取决于要索引的数据量。最好在服务器负荷不重时执行这一操作。增大 `batch_size` 配置参数的值也可加快该过程的执行速度。有关详细信息，请参见第 76 页的“`batch_size`”。

注释 不要重命名索引； Verity 集合不可重命名。

创建文本索引时指定多列

当在两列或更多列上创建文本索引时，文本索引中的每一列都放入一个单独的文档区域中。该区域的名称就是列名。例如，若要在 KRAZYKAT 上为 pubs2 中 blurbs 表的 copy 列和 au_id 列创建文本索引和一个名为 i_blurbs 的索引表，请输入：

```
sp_create_text_index "KRAZYKAT", "i_blurbs", "blurbs", " ", "copy", "au_id"
```

`sp_create_text_index` 在文本索引中创建名称分别为“copy”和“au_id”的两个区域。当向 i_blurbs 文本索引发出一个查询时，搜索范围将包括 copy 和 au_id 列。不过，可使用 in 运算符指定一个文档区域，将搜索范围限制到某一特定列（有关详细信息，请参见第 50 页的“in”）。

授予对文本索引代理表的权限

通过 `sp_create_text_index` 创建文本索引后，管理员必须明确地向将要查询文本索引的用户或组授予对文本索引代理表的 `select` 权限。

用户必须对文本索引源中的所有列都有 `select` 权限，才能被授予对文本索引的 `select` 权限。如果用户对文本索引源中的任何列都没有 `select` 权限，管理员就不能向该用户授予对文本索引的 `select` 权限。

如果文本索引源表中有加密列，则管理员应仅向对加密列有 `decrypt` 权限的用户授予对文本索引的 `select` 权限。如果用户对加密列有 `select` 权限，但没有 `decrypt` 权限，管理员就不应向该用户授予对文本索引的 `select` 权限。

使数据库联机以进行全文本搜索

使用增强型全文本搜索引擎时，数据库会在 `auto_online` 配置参数设置为 1 时自动联机。

当使数据库联机时，增强型全文本搜索引擎会对内部的 Verity 结构进行初始化并确认 Verity 集合存在。

如果数据库没有自动联机，可使用 `sp_text_online` 使数据库联机以进行全文本搜索。例如，若要在名为 `KRAZYKAT` 的增强型全文本搜索引擎中的 `blurbs` 表上发出全文本搜索之前，使 `pubs2` 数据库联机，请输入：

```
sp_text_online KRAZYKAT, pubs2
```

将出现如下消息：

```
Database 'pubs2' is now online
```

现在 `pubs2` 数据库可用来执行全文本搜索了。

有关详细信息，请参见第 146 页的“`sp_text_online`”。

将更改传播到文本索引

当在源表中 `insert`、`update` 或 `delete` 数据时，文本索引不会自动更新。更新数据后，可运行 `sp_refresh_text_index` 将更改记录到 `text_events` 表中，再运行 `sp_text_notify` 以通知增强型全文本搜索引擎需要处理更改。然后，增强型全文本搜索引擎就会连接到 Adaptive Server，读取 `text_events` 表中的条目，确定哪些索引、表和行受到了影响，并更新相应的集合。

有关这些系统过程的详细信息，请参见第 135 页的“`sp_refresh_text_index`”和第 145 页的“`sp_text_notify`”。

为使每次 `insert`、`update` 或 `delete` 操作后，`sp_refresh_text_index` 都自动运行，可以在源表中创建触发器，方法如下：

- 创建在每次执行 `delete` 操作后都运行 `sp_refresh_text_index` 的触发器。
- 创建在每次执行 `insert` 操作后都运行 `sp_refresh_text_index` 的触发器。
- 创建在每次对索引列执行 `update` 操作后都运行 `sp_refresh_text_index` 的触发器。

当使用 `writetext` 更新 `text` 列时，不会引发触发器。使 `sp_refresh_text_index` 在 `writetext` 操作后自动运行：

- 设置一个非 `text` 列并在每一次 `writetext` 操作后对该列执行 `update`。
- 在非 `text` 列上创建触发器以运行 `sp_refresh_text_index`。因为增强型全文本搜索引擎会在您发出 `sp_text_notify` 时重新插入整个行，所以 `text` 列的更新将会传播到文本索引。

有关上述每个触发器的示例，请参见 `$$SYBASE/$$SYBASE_FTS/sample/scripts` 目录中的示例脚本 `sample_text_main.sql`。

复制文本索引

若要复制具有文本索引的表，请遵循以下原则：

- 在目标数据库中创建表定义。
- 如果 `text_events` 表尚不存在，请运行 `installevnt` 脚本在目标数据库中创建 `text_events` 表（请参见第 14 页的“运行 `installevnt` 脚本”）。
- 运行 `sp_create_text_index`，在目标数据库中的空表上创建文本索引（请参见第 18 页的“创建文本索引和索引表”）。
- 创建一些触发器，以便每当在表中 `insert`、`update` 或 `delete` 数据时，都会运行 `sp_create_text_index` 以便向 `text_events` 表中 `insert` 相应条目（请参见第 20 页的“将更改传播到文本索引”）。
- 在 Replication Server 中创建复制定义，以便将源表中的所有数据复制到目标表。有关详细信息，请参见《Replication Server 管理指南》。
- 运行 `sp_text_notify` 以 `update` 文本索引；定期运行 `sp_text_notify` 以处理目标表发生的更改（请参见第 20 页的“将更改传播到文本索引”）。

注释 只要执行 `writetext` 命令，就必须向非 `text` 列发出 `update`。该操作可确保引发向 `text_events` 表插入数据的触发器。

示例：启用新的数据库以进行文本搜索

本示例描述了在 `movies` 数据库中 `reviews` 表的 `plot` 列上创建 `text` 索引的步骤。示例中假设：

- 已经在 `MYSVR` 服务器上的一个名为 `movies` 的新数据库中创建了 `reviews` 表。
- `reviews` 表有一个需要索引的 `plot` 列。
- `Adaptive Server` 与名为 `MYTXTSVR` 的增强型全文本搜索引擎已配置为相互连接

步骤 1：确认 `text_events` 表是否存在

每个包含由文本索引引用的表的数据库都必须包含一个 `text_events` 表，以记录索引列的插入、更新和删除。

如果 `text_events` 表存在于 `model` 数据库中，则它也将存在于所有新数据库中。如果 `text_events` 表不在 `model` 数据库中，请运行 `installevent` 脚本在新数据库中安装 `text_events` 表。例如，要在 `movies` 数据库中安装 `text_events` 表，请执行以下操作：

- 将 `installevent` 脚本另存为 `installeventmovies`。
- 编辑脚本，将所有 `model` 单词替换为单词 `movies`。
- 按如下所示运行脚本：

```
isql -Usa -P -SMYSVR -i
$SYBASE/$SYBASE_FTS/scripts/installeventmovies
```

有关安装 `text_events` 表的信息，请参见第 14 页的“[运行 `installevent` 脚本](#)”。

步骤 2：检查 `IDENTITY` 列或主键

每个源表都必须包含 `IDENTITY` 列或主键，该列或主键唯一标识每一行并提供连接索引表和源表的方法。

例如，若要向 `reviews` 表添加 `IDENTITY` 列，请输入：

```
alter table reviews add id numeric(10,0) identity
```

有关详细信息，请参见第 17 页的“[向源表添加 `IDENTITY` 列](#)”。

步骤 3: 在 IDENTITY 列上创建唯一索引

此步骤是可选的。为改善性能，Sybase 建议您创建仅包含 IDENTITY 列的唯一索引。例如，若要在上一步所创建的 IDENTITY 列上创建一个名为 `reviews_id` 的唯一索引，请发出以下命令：

```
create unique index reviews_id on reviews(id)
```

有关创建唯一索引的详细信息，请参见《Transact-SQL 用户指南》的第 11 章“创建表的索引”。

步骤 4: 创建文本索引和索引表

只有在对用户数据库中的源表进行索引后才可以执行全文本搜索。例如，若要为 `reviews` 表的 `plot` 列创建文本索引和一个名为 `reviews_idx` 的索引表，请输入：

```
sp_create_text_index "MYTXTSVR", "reviews_idx",  
"reviews", " ", "plot"
```

现在 `reviews` 表可用来运行全文本搜索了。

有关详细信息，请参见第 129 页的“[sp_create_text_index](#)”。

步骤 5: 使数据库联机以进行全文本搜索

若要使名为 MYTXTSVR 的增强型全文本搜索引擎的 `movies` 数据库联机，请输入：

```
sp_text_online MYTXTSVR, movies
```

注释 如果已将 `auto_online` 设置为“1”，可省略这一步。

有关详细信息，请参见第 146 页的 [sp_text_online](#)。

对欧元符号进行索引

如果遵循以下配置规则，则可以正确地索引和返回欧元符号。Adaptive Server 必须安装有 `utf8` 字符集。增强型全文本搜索引擎必须已将 `vdkLanguage` 设置为 `<语言>x`，并将 `vdkCharset` 保留为空白。例如：

```
ASE 12.5.x charset = utf8  
EFTS 12.5.x vdkLanguage = englishx  
EFTS 12.5.x vdkCharset =
```


设置 Verity 功能

本章介绍在利用某些 Verity 功能编写查询时所需要的设置。

主题	页码
启用按示例查询、摘要和集群	25
设置用作排序规范的列	28
对包含标记的文本使用过滤器	30
创建自定义同义词库	32
创建主题	35

启用按示例查询、摘要和集群

style.prm 文件指定为支持以下功能而需要在文本索引中包含的其它数据：

- 按示例查询 — 检索与某个短语相似的文档（有关详细信息，请参见第 51 页的“like”）。

注释 文本索引需要其它数据来支持 like 运算符的按示例查询规范中的短语。如果在按示例查询规范中使用文档，则不需要其它数据。

- 摘要 — 返回文档的摘要，而不是整个文档（有关详细信息，请参见第 45 页的“使用 summary 列摘要文档”）。
- 集群 — 按子主题在结果集中对文档进行分组。有关详细信息，请参见第 45 页的“使用伪列请求集群结果集”。

可以通过编辑主 *style.prm* 文件来启用所有文本索引的功能，或者通过编辑相应的 *style.prm* 文件来启用各个文本索引的功能。下面将介绍这两种方法。

按示例查询和集群

若要在按示例查询规范中使用短语或者要使用集群，必须在建立索引时启用对文档功能矢量的生成。为此，需要将 *style.prm* 文件中的以下行撤消注释：

```
$define DOC-FEATURES "TF"
```

摘要

若要为摘要功能配置增强型全文本搜索引擎，请将 *style.prm* 文件中的下列以 “*#\$define*” 开头的行之一撤消注释：

```
# The example below stores the best three sentences of
# the document, but not more than 255 bytes.
#$define DOC-SUMMARIES  "XS MaxSents 3 MaxBytes 255"
# The example below stores the first four sentences of
# the document, but not more than 255 bytes.
#$define DOC-SUMMARIES  "LS MaxSents 4 MaxBytes 255"
# The example below stores the first 150 bytes of
# the document, with whitespace compressed.
#$define DOC-SUMMARIES  "LB MaxBytes 150"
```

这些行中的每一行都反映了摘要的不同级别。通过更改这些行末尾的数字，可以指定希望增强型全文本搜索引擎显示的数据的字节数。例如，如果仅摘要数据的前 233 个字节，可按如下方式编辑脚本：

```
$define DOC-SUMMARIES  "LS MaxSents 4 MaxBytes 233"
```

显示的最大字节数为 255。任何大于 255 的数均被截断为 255。

编辑主 *style.prm* 文件

注释 从 EFTS 12.5.2 版本开始，主 *style.prm* 文件的位置已经移动。对 *style.prm* 文件（位于 *\$\$SYBASE/\$\$SYBASE_FTS/verity/common/style* 中）所做的编辑将被忽略。新位置按如下方式指定。

主 *style.prm* 文件位于 *\$\$SYBASE/\$\$SYBASE_FTS/verity/common/styles/txtsvr* 中。它包含增强型全文本搜索引擎缺省的样式参数。可以通过编辑该文件来配置增强型全文本搜索引擎，以便使所有创建了文本索引的表都允许在按示例查询规范或摘要功能中使用集群和文本字符。按上文所述撤消对相应行的注释。

注释 如果文本索引已存在，则必须重新创建文本索引，该文本索引启用具有如在第 27 页的“编辑单个 *style.prm* 文件”中描述的功能。

编辑单个 *style.prm* 文件

执行以下步骤可以配置增强型全文本搜索引擎，以便单个文本索引允许在按示例查询规范或摘要中使用集群和文本字符：

- 1 使用 `sp_create_text_index` 创建文本索引。在 `option_string` 参数中使用“empty”一词，这样可以为文本索引创建 *style.prm* 文件，但在 Verity 集合中不填充数据。例如，若要启用 `blurbs` 表的 `copy` 列的集群，请使用以下语法：

```
sp_create_text_index "KRAZYKAT", "i_blurbs", "blurbs", "empty", "copy"
```

注释 如果已存在文本索引，请忽略这一步。不需要再次创建文本索引。

- 2 使用 `sp_drop_text_index` 删除与正编辑的 *style.prm* 文件相关联的文本索引。

例如，若要删除在步骤 1 中创建的文本索引，请输入：

```
sp_drop_text_index "blurbs.i_blurbs"
```

- 3 编辑用于文本索引的 *style.prm* 文件。现有集合的 *style.prm* 文件位于 `$$SYBASE/$$SYBASE_FTS/collections/db.owner.index/style` 中。

例如，如果在 `pubs2` 数据库上创建了名为 `i_blurbs` 的文本索引，则到这些文件的完整路径应如下所示，其中 `db.owner.index` 表示数据库、数据库所有者和使用 `sp_create_text_index` 创建的索引：
`$$SYBASE/$$SYBASE_FTS/collections/pubs2.dbo.i_blurbs/style`

- 4 按上文所述撤消对相应行的注释。例如，若要启用集群，请撤消对下面一行的注释：

```
$define DOC-FEATURES "TF"
```

- 5 重新创建在步骤 2 中删除的文本索引。例如，若要重新创建 `i_blurbs` 文本索引，请输入：

```
sp_create_text_index "KRAZYKAT", "i_blurbs", "blurbs", "", "copy"
```

设置用作排序规范的列

在按特定列排序之前，必须修改 *style.vgw* 和 *style.ufl* 文件。（有关在排序规范中包含列的信息，请参见第 44 页的“使用 *sort_by* 列指定排序顺序”。）这两个文件位于以下位置，其中 *db.owner.index* 表示数据库、数据库所有者和使用 *sp_create_text_index* 创建的索引：
\$\$SYBASE/\$\$SYBASE_FTS/collections/db.owner.index/style

例如，如果在 *pubs2* 数据库上创建了名为 *i_blurbs* 的文本索引，则到这些文件的完整路径应为
\$\$SYBASE/\$\$SYBASE_FTS/collections/pubs2.dbo.i_blurbs/style

若要编辑 *style.vgw* and *style.ufl* 文件，请执行以下步骤：

- 1 删除包含将要为其添加定义的列的文本索引。删除文本索引并不会删除集合目录。

例如，若要在 *blurbs* 表中添加 *copy* 列的定义，请使用如下命令删除文本索引：

```
sp_drop_text_index i_blurbs
```

- 2 编辑 *style.vgw* 文件。在下行的后面：

```
dda "SybaseTextServer"
```

为要定义的列添加一个条目。语法如下所示（其中，*column_number* 是要定义的列的编号）：

```
table: DOCUMENTS
{
    copy: fcolumn_number copy_column_number
}
```

列序号从 0 开始；如果要对第一列排序，请指定“f0”；要对第二列排序，指定“f1”；要对第三列排序，指定“f2”，依此类推。

例如，若要定义表的第一列，命令语法为：

```
table: DOCUMENTS
{
    copy: f0 copy_f0
}
```

因此，*style.vgw* 文件将类似于下面的内容：

```
#
#       Sybase Text Server Gateway
#
$control: 1
gateway:
{
    dda:    "SybaseTextServer"
    {
        copy: f0 copy_f0
    }
}
```

- 3 通过添加名为 **fts** 的数据表的列定义，对 *style.ufl* 文件进行编辑。语法为：

```
data-table:    fts
{
    fixwidth:    copy_fcolumn_number precision datatype
}
```

列序号从 0 开始；如果要对第一列排序，请指定“f0”；要对第二列排序，指定“f1”；要对第三列排序，指定“f2”，依此类推。例如，若要为表的第一列添加一个定义，规定该列的精度为 4，数据类型为 **date**，请输入：

```
data-table: fts
{
    fixwidth:    copy_f0    4    date
}
```

同样，若要为表的第二列添加一个定义，规定该列的精度为 10，数据类型为 **character**，请输入：

```
data-table: fts
{
    fixwidth:    copy_f1    10    text
}
```

- 4 使用 **sp_create_text_index** 重新创建索引。

对包含标记的文本使用过滤器

若要对包含标记（如 HTML 或 Post Script）的文档执行精确搜索，文本索引必须使用过滤器来剔除标记。增强型全文本搜索引擎提供了用于各种文档类型（Microsoft Word、FrameMaker、WordPerfect、SGML、HTML 及其它类型）的过滤器。

创建要使用过滤器的文本索引时，文档中对应于每种类型标记的数据均被放到各自的文档区域中。例如，如果有名为“chapter”的标记，则所有的章节名均被放到一个文档区域中。您可以发布搜索整个文档的查询，或仅搜索“chapter”区域中的数据（有关详细信息，请参见第 50 页的“in”）。

若要创建使用过滤器的文本索引，请修改该文本索引对应的 *style.dft* 文件：

- 1 使用 `sp_create_text_index` 创建文本索引。在 *option_string* 参数中使用“empty”一词，这样可以为文本索引创建 *style.dft* 文件，但在 Verity 集合中不填充数据。例如，若要创建 `blurbs` 表的 `copy` 列的文本索引，请使用如下语法：

```
sp_create_text_index "KRAZYKAT", "i_blurbs", "blurbs", "empty", "copy"
```

- 2 删除在步骤 1 中创建的文本索引。这将会删除该文本索引，但不删除 *style.dft* 文件。例如，可使用以下命令删除 `i_blurbs` 文本索引：

```
sp_drop_text_index i_blurbs
```

- 3 编辑 *style.dft* 文件。*style.dft* 文件位于 `$$SYBASE/$SYBASE_FTS/collections/db.owner.index/style` 目录中，其中 `db.owner.index` 表示数据库、数据库所有者和使用 `sp_create_text_index` 创建的索引。例如，如果在 `pubs2` 数据库上创建了名为 `i_blurbs` 的文本索引，则到 *style.dft* 文件的完整路径应为 `$$SYBASE/$SYBASE_FTS/collections/pubs2.dbo.i_blurbs/style`

在下行的后面：

```
field: f0
```

添加使用过滤器的语法：

- 对于 SGML 文档，使用：

```
/filter="zone -nocharmap"
```

- 对于 HTML 文档，使用：

```
/filter="zone -html -nocharmap"
```


对所有文档类型均可使用以下语法：

```
/filter="universal"
```

例如，SGML 文档的 *style.dft* 文件将类似于：

```
$control: 1
dft:
{
    field: f0
        /filter="zone -nocharmap"
    field: f1
    field: f2
    .
    .
    field: f15
}
```

SGML 文档的 *style.dft* 文件将类似于：

```
$control: 1
dft:
{
    field: f0
        /filter="universal"
    field: f1
    field: f2
    .
    .
    field: f15
}
```

注释 可以使用 `getsend` 将文档数据装载到数据库中。`getsend` 采用以下参数：`database`、`table`、`column` 和 `row id`。为希望插入的每一行文本的 `rowid` 插入一个空值。`getsend` 必须插入一个 `image` 列才能使用过滤功能。有关 `getsend` 的详细信息，请参考 `$$SYBASE/$SYBASE_FTS/sample/source` 目录中的 `README.TXT` 文件和 `getsend.c` 文件。

4 使用 `sp_create_text_index` 重新创建索引。例如：

```
sp_create_text_index "KRAZYKAT", "i_blurbs", "blurbs", "", "copy"
```

创建自定义同义词库

Verity thesaurus 运算符将搜索扩展到包括指定的单词及其同义词（有关使用 thesaurus 运算符的信息，请参见第 54 页的“thesaurus”）。可创建一个自定义的同义词库，并在其中包含应用程序特定的同义词以代替缺省的同义词库。

例如，缺省的英语同义词库中包含以下单词作为“money”的同义词：“cash”、“currency”、“lucre”、“wampum”和“greenbacks”。您可以创建一个自定义同义词库，在其中包含另一组单词作为“money”的同义词，如：“bid”、“tokens”、“credit”、“asset”和“verbal offer”。

创建自定义同义词库：

- 1 编制用于您的应用程序的同义词列表。该列表可帮助检查缺省的同义词库（请参见第 33 页的“检查缺省同义词库（可选项）”）。
- 2 创建一个包含您为自定义同义词库定义的同义词的控制文件（请参见第 33 页的“创建控制文件”）。
- 3 使用 mksyd 实用程序创建自定义同义词库（请参见第 34 页的“创建同义词库”）。mksyd 实用程序位于 `$$SYBASE/$$SYBASE_FTS/verity/<verity_platform_directory>/bin` 中。创建过程将使用控制文件作为输入。
- 4 用自定义同义词库代替缺省的同义词库（请参见第 35 页的“用自定义同义词库替换缺省同义词库”）。

有关“自定义同义词库支持”和 mksyd 实用程序的详细信息，请参见位于 <http://www.verity.com> 的 Verity 网站。

以下两个示例文件说明了如何设置和使用自定义同义词库：

- `sample_text_thesaurus.ctl` 是示例控制文件。
- `sample_text_thesaurus.sql` 向示例控制文件中定义的自定义同义词库发出查询。

这些文件在 `$$SYBASE/$$SYBASE_FTS/sample/scripts` 目录下。

检查缺省同义词库（可选项）

控制文件包含同义词库的所有同义词定义。若要检查缺省的同义词库，请使用 `mksyd` 实用程序创建它的控制文件。`mksyd` 实用程序位于 `$SYBASE/SYBASE_FTS/verity/<verity_platform_directory>/bin` 中。

使用以下语法：

```
mksyd -dump -syd
$SYBASE/$SYBASE_FTS/verity/common/vdkLanguage/vdk20.syd
-f work_location/control_file.ctl
```

其中：

- `vdkLanguage` 是 `vdkLanguage` 配置参数的值（例如，“english”）。
- `work_location` 是要放入控制文件的目录。
- `control_file` 是从缺省同义词库创建的控制文件的名称。

若要查看缺省同义词列表，请检查所创建的控制文件 (`control_file.ctl`)。

创建控制文件

创建一个包含自定义同义词库中新同义词的控制文件。控制文件是采用结构化格式的 ASCII 文本文件。使用文本编辑器（例如 `vi` 或 `emacs`）来执行以下操作：

- 编辑来自缺省同义词库的控制文件，并向现有同义词库中添加新的同义词（请参见第 33 页的“检查缺省同义词库（可选项）”，或者
- 创建一个仅包含您的同义词的新控制文件。

控制文件语法

控制文件在 `synonyms:` 语句中包含同义词列表定义。例如，下面是一个名为 `colors.ctl` 的控制文件：

```
$control: 1
synonyms:
{
list: "red, ruby, scarlet, fuchsia,\
magenta"
list: "electric blue <or> azure"
/keys = "lapis"
}
$$
```

synonyms: 语句包括:

- **list:** 关键字, 用以指定同义词列表的开始位置。列表中的同义词既可以采用查询格式, 也可以是以逗号分隔的单词或短语的列表。
- 每个 **list:** 都可以选择使用一个 **/keys** 修饰符, 用以指定一个键或多个由逗号分隔的键。在上面的示例中, 第一个 “list” 中没有指定任何键。这意味着, 当在同义词库中查询 “red”、“ruby”、“scarlet”、“fuchsia” 或 “magenta” 时, 均会找到该列表。第二个 “list” 使用 **/keys** 修饰符指定了一个键。这意味着, 仅当指定 `<thesaurus>lapis` 时, 该列表中的单词或短语才满足查询条件。

注释 如果使用 **emacs** 建立同义词列表且所有列表都超过了一行, 请关闭 **auto-fill** 模式。如果将列表分成多个行, 请在每行的结尾包括一个反斜杠 (`\`), 以便将这些行视为一个列表。

有关控制文件的更为复杂的示例, 请访问 **Verity Web** 站点。

创建同义词库

mksyd 实用程序将控制文件作为输入来创建自定义同义词库。该实用程序位于:

```
$$SYBASE/$$SYBASE_FTS/verity/<verity_platform_directory>/bin
```

从 **bin** 目录运行或定义一个别名运行 **mksyd**。在任意工作目录中创建自定义同义词库。

创建自定义同义词库的 **mksyd** 语法是:

```
mksyd -f control_file.ctl -syd custom_thesaurus.syd
```

其中:

- **control_file** — 是在前面一节中创建的控制文件的名称。
- **custom_thesaurus** — 是要创建的自定义同义词库的名称。

例如, 若要执行 **mksyd** 实用程序, 以读取上面定义的示例控制文件并将输出保存到一个工作目录中, 请使用:

```
mksyd -f /usr/u/sybase/dba/thesaurus/colors.ctl  
-syd /usr/u/sybase/dba/thesaurus/custom.syd
```

用自定义同义词库替换缺省同义词库

名为 `vdk20.syd` 的缺省同义词库位于 `$$SYBASE/$$SYBASE_FTS/verity/common/vdkLanguage` 中，其中，`vdkLanguage` 是 `vdkLanguage` 配置参数的值（例如，英文目录是 `$$SYBASE/$$SYBASE_FTS/verity/common/english`）。运行时从该位置读取的每个应用程序和用户均使用该同义词库。用自定义同义词库替换缺省的同义词库：

- 1 在用自定义同义词库替换缺省同义词库之前，备份缺省同义词库。
例如：

```
mv /$SYBASE/$SYBASE_FTS/verity/common/english/vdk20.syd default.syd
```

- 2 用自定义同义词库替换 `vdk20.syd` 文件。例如：

```
cp custom.syd /$SYBASE/$SYBASE_FTS/verity/common/english/vdk20.syd
```

- 3 重新启动对控制文件 (`control_file.ctl`) 的检查；不需要更改配置文件。启动对控制文件 (`control_file.ctl`) 的检查时（不是执行查询时），将从该位置读取同义词库。

使用 `thesaurus` 运算符的查询现在将使用自定义同义词库。

创建主题

本节提供对 Verity 主题的简要概述。该主题在第 8 章“[Verity 主题](#)”中进行了详细讨论。

主题是与某个概念或科目范围相关的一组信息。有了主题定义以后，用户便可以根据主题执行搜索，而不必用复杂的语法编写查询了。

用户可创建由单词和短语、Verity 运算符和修饰符以及权值等组合而成的主题。此后，任何用户就可以查询主题。

在创建主题之前，应确定应用程序的需要，并为命名约定和以下各项的位置建立标准：

- 大纲文件 — 包含主题定义。每个主题均有自己的大纲文件。
- 主题组目录 — 包含汇编主题。每个主题均有自己的主题组目录。
- 知识库映射文件 — 包含指向主题组目录的指针。

若要实现主题，请执行下列步骤：

- 1 创建一个或多个大纲输入文件以定义主题（请参见第 36 页的“创建大纲文件”）。每个大纲文件都用于填充一个主题组。
- 2 使用 `mktopics` 实用程序创建并填充主题组目录（请参见第 37 页的“创建主题组目录”）。`mktopics` 实用程序位于 `$$SYBASE/$SYBASE_FTS/verity/<verity_platform_directory>/bin` 中。每个主题组目录均基于一个主题大纲输入文件进行填充。
- 3 创建知识库映射，指定一个或多个主题组目录的位置（请参见第 38 页的“创建知识库映射”）。
- 4 将 `knowledge_base` 配置参数设置为指向知识库映射的位置（请参见第 38 页的“定义知识库映射的位置”）。
- 5 对所定义的主题执行查询。

下面的示例文件说明了主题的功能：

- `sample_text_topics.otl` 是示例大纲文件。
- `sample_text_topics.kbm` 是示例知识库映射。
- `sample_text_topics.sql` 使用所定义的主题发出查询。

这些文件在 `$$SYBASE/$SYBASE_FTS/sample/scripts` 目录下。

创建大纲文件

当使用主题发出查询时，主题大纲文件将指定搜索引擎使用的单词和短语、Verity 运算符和修正符及权值的所有组合。大纲文件是采用结构化格式的 ASCII 文本文件。

例如，以下大纲文件定义主题“saint-bernard”：

```
$control: 1
saint-bernard <accrue>
*0.80 "Saint Bernard"
*0.80 "St. Bernard"
* "working dogs"
* "large dogs"
* "European breeds"
$$
```

当发布指定主题 “saint-bernard” 的查询时，增强型全文本搜索引擎将：

- 返回包含一个或多个以下短语的文档：“Saint Bernard”、“St. Bernard”、“working dogs”、“large dogs”和“European breeds”
- 为包含短语“Saint Bernard”或“St. Bernard”的文档评定比包含短语“working dogs”、“large dogs”或“European breeds”的文档更高的得分

该示例是一个最基本的主题定义。大纲可以通过以下方式建立更复杂的关系：

- 多级子主题
- Verity 运算符的组合（该示例使用 `accrue`）
- Verity 修饰符

注释 在 Windows NT 中，可使用 Verity 提供的 Verity 智能分类器产品的图形用户界面来创建主题大纲。如果使用智能分类器，它将自动创建主题组目录，您可以按照第 38 页的“创建知识库映射”的说明继续设置主题。

创建主题组目录

使用 `mktopics` 实用程序可创建并填充主题组目录。此实用程序位于 `$$SYBASE/$SYBASE_FTS/verity/<verity_platform_directory>/bin` 中。

从该 `bin` 目录运行或定义一个别名运行 `mktopics`。可在任何工作目录下创建一个或多个主题组目录。

`mktopics` 的语法如下：

```
mktopics -outline outline_file.otl -topicset topic_set_directory
```

其中：

- *outline_file* — 是您在第 36 页的“创建大纲文件”中创建的大纲文件的名称。
- *topic_set_directory* — 是要创建的主题组目录的名称。

例如，若要执行 `mktopics` 实用程序，以读取上面定义的 `saint-bernard.otl` 文件并将输出保存到一个工作目录中，请使用以下语法：

```
mktopics -outline
/usr/u/sybase/topic_outlines/saint-bernard.otl
-topicset /usr/u/sybase/topic_sets/
saint-bernard_topic
```

创建知识库映射

知识库映射 可指定一个或多个主题组目录的位置。创建 ASCII 知识库映射文件，并在其中定义指向主题组的完全限定目录路径。

例如，下面的知识库映射文件显示了如何在映射中列出多个知识库。第一个条目标识由前面的 `mktopics` 创建的主题组目录。

```
$control:1
kbases:
{
kb:
/kb-path = /usr/u/sybase/topic_sets/saint-bernard_topic
kb:
/kb-path = /usr/u/sybase/topic_sets/another_topic
}
```

定义知识库映射的位置

设置 `knowledge_base` 配置参数，以指向知识库映射的位置。例如：

```
sp_text_configure KRAZYKAT, 'knowledge_base',
'/usr/u/sybase/topic_sets/sample_text_topics.kbm'
```

`knowledge_base` 配置参数是静态的，必须重新启动增强型全文本搜索引擎才能使定义生效。

对定义的主题执行查询

现在，可以用所定义的主题代替复杂查询来执行查询了。例如，在创建“saint-bernard”主题之前，必须使用以下语法：

```
...where i.index_any = "<accrue> ([80]Saint Bernard, [80]St. Bernard, working dogs, large dogs, European breeds)"
```

查找满足下列条件的文档：

- 包含一个或多个以下短语：“Saint Bernard”、“St. Bernard”、“working dogs”、“large dogs”和“European breeds”
- 为包含短语“Saint Bernard”或“St. Bernard”的文档评定比包含短语“working dogs”、“large dogs”或“European breeds”的文档更高的得分

在创建主题“saint-bernard”之后，则可使用以下语法：

```
...where i.index_any = "<topic>saint-bernard"
```

或：

```
...where i.index_any = "saint bernard"
```

注释 如果在查询表达式中输入一个单词，增强型全文本搜索引擎会尝试将它与一个主题名进行匹配。如果在查询表达式中输入一个短语，增强型全文本搜索引擎会用连字符 (-) 代替空格，并尝试将它与一个主题名进行匹配。例如，增强型全文本搜索引擎会将“saint bernard”与主题“saint-bernard”进行匹配。

有关在查询中使用主题的示例，请参见 *sample_text_topics.sql* 文件。

主题疑难解答

如果 `knowledge_base` 配置参数指定了不存在的知识库映射文件，增强型全文本搜索引擎将不能用 Verity 启动会话，而且服务器将不启动。如果映射文件存在但包含无效条目，则 Verity 会在启动时发出警告消息。您可以通过编辑 `$$SYBASE` 目录中的 `<textserver>.cfg` 文件更正错误。您可以更正路径信息并更改以“`knowledge_base=`”开始的行。

编写全文本搜索查询

本章介绍全文本搜索中可包括的伪列、搜索运算符和修饰符。

主题	页码
全文本搜索查询的组件	41
索引表中的伪列	42
全文本搜索运算符	47
运算符修饰符	57

全文本搜索查询的组件

若要编写全文本搜索查询，请输入查询参数，作为 Adaptive Server select 语句的一部分，然后由增强型全文本搜索引擎进行搜索。

select 语句需要：

- 一个 where 子句，将 Verity 语言查询分配给 index_any 伪列
- 用于进一步定义搜索参数的伪列（可选）
- 源表中的 IDENTITY 列或主键与索引表中的 id 列之间的连接

例如，要返回 blurbs 表的 copy 列中经常出现“software”一词的 10 个文档，请输入：

```
select t1.score, t2.copy
from i_blurbs t1, blurbs t2
where t1.id=t2.id
and t1.index_any = "<many> <word> software"
and t1.max_docs = 10
```

Adaptive Server 将 Verity 查询传递到增强型全文本搜索引擎来进行搜索。请参见第 8 页的“全文本搜索的工作方式”。

缺省行为

增强型全文本搜索引擎查询的缺省语法或简单语法一般符合下列原则：

- 1 搜索区分大小写。
- 2 STEM 运算符适用于搜索单词。
- 3 应用 MANY 修饰符。
- 4 ACCRUE 运算符在父级激活。

索引表中的伪列

伪列 是索引表中的列，它定义搜索的参数并提供对结果数据的访问。请参见第 5 页的“[index 表](#)”。这些列只有在查询环境中才有效；也就是说，列中的信息只在查询期间有效。如果随后的查询包含不同的参数集，伪列就包含不同组数值。

索引表中的每个伪列描述一个不同的搜索属性。例如，如果指出 `score` 列，则查询只显示属于所定义参数范围内的结果集。例如，以下查询只显示 `score` 值大于 90 的结果：

```
index_table_name.score > 90
```

其它伪列（如 `highlight`）用来检索由 Verity 为特定文档生成的数据。[表 5-1](#) 介绍由增强型全文本搜索引擎维护的伪列。

表 5-1: 增强型全文本搜索引擎的伪列

伪列名	说明	数据类型	长度 (单位为 字节)
cluster_number	包含集群，该行是该集群的一部分。集群编号从 1 开始。只能在查询的 <code>select</code> 子句中使用 <code>cluster_number</code> 列。	int	4
cluster_keywords	包含 Verity 用来建立集群的关键字。只能在查询的 <code>select</code> 子句中使用 <code>cluster_keywords</code> 。	varchar	255
highlight	偏移文档中来自查询的所有单词。只能在查询的 <code>select</code> 子句中使用 <code>highlight</code> 。	text	16
id	在集合内唯一标识一个文档。用来与源表的 IDENTITY 列连接。可以在查询的 <code>select</code> 子句或 <code>where</code> 子句中使用 <code>id</code> 。	numeric	6
index_any	为增强型全文本搜索引擎提供 Verity 语言查询。只能在 <code>where</code> 子句中使用 <code>index_any</code> 。尽管伪列定义为 <code>char(255)</code> ，但 <code>index_any</code> 子句的最大长度为 16000。	varchar	255

伪列名	说明	数据类型	长度 (单位为 字节)
max_docs	根据缺省的排序顺序，将结果限制到前 n 个文档。在一个集群结果集中，将结果限制到每个集群中的前 n 个文档。只能在 <code>where</code> 子句中使用 <code>max_docs</code> 。	int	4
score	搜索字符串与索引列之间相互关系的规范化度量。与特定文档相关联的 <code>score</code> 仅在用来检索文档的查询的引用中才有意义。可以在 <code>select</code> 子句或 <code>where</code> 子句中使用 <code>score</code> 。	int	4
sort_by	指定排序顺序，结果集按此顺序返回。 增强型全文本搜索引擎允许在 <code>sort_by</code> 列中最多有 16 个排序规范。 只能在 <code>where</code> 子句中使用 <code>sort_by</code> 。	varchar	35
summary	选择摘要数据。只能在查询的 <code>select</code> 子句中使用 <code>summary</code> 列。	varchar	255
total_docs	包含符合搜索条件的文档总数。	int	4

接下来的部分将介绍伪列的功能。

使用 `score` 列按相关性排列搜索结果

相关性排列 是增强型全文本搜索引擎的一个功能，用来给 `score` 参数分配一个值，以指示文档符合查询条件的程度。`score` 计算依赖于查询中使用的搜索运算符。请参见第 49 页的“使用 `Verity` 运算符”。文档越符合查询条件，它的 `score` 值就越高。

例如，如果搜索包含单词“rain”的文档，出现 12 次“rain”的文档的 `score` 值就比出现 6 次“rain”的文档高。

如果将查询中的 `score` 值设置得很高（如 90），那么结果集就会限定为其 `score` 值高于这个数值的文档。

注释 `Verity` 的 `score` 值使用小数；而 `Sybase` 使用整数。例如，如果 `Verity` 报告的 `score` 值为 .85，`Sybase` 对此相同值将报告 85。

例如，下面的查询搜索包含单词“`raconteur`”或“`Paris`”或同时包含这两个单词的文档，并且 `score` 要大于或等于 90：

```
select t1.score, t2.copy
from i_blurbs t1, blurbs t2
where t1.id=t2.id and t1.score > 90
and t1.index_any = "<accrue>(raconteur, Paris)"
score      copy
-----
```

(0 rows affected)

此查询未能找到任何包含单词 “raconteur” 或 “Paris” 并且得分大于 90 的文档。但是如果查询中的 *score* 值降到 39，就会找到在 *blurbs* 表中有一个文档涉及到 “raconteur” 或 “Paris”：

```
select t1.score, t2.copy
from i_blurbs t1, blurbs t2
where t1.id=t2.id and t1.score > 39
and t1.index_any = "<accrue>(raconteur, Paris)"
score      copy
-----
40         A chef's chef and a raconteur's raconteur, Reginald
          Blotchet-Halls calls London his second home. "Th' palace
          . . .
```

使用 *sort_by* 列指定排序顺序

排序顺序指定用来对结果集中的数据进行排序的归类序列。缺省排序顺序由 *sort_order* 配置参数设置。请参见第 67 页的 “设置缺省排序顺序”。支持不区分大小写的排序顺序。

使用 *sort_by* 伪列可返回一个采用非缺省排序顺序的结果集。最多可以在 *sort_by* 伪列中指定 16 个排序规范。

表 5-2 列出了 *sort_by* 伪列的值。

表 5-2: *sort_by* 伪列的值

值	返回
<i>fts_score</i> desc	按 <i>score</i> 伪列降序排列的结果集。
<i>fts_score</i> asc	按 <i>score</i> 伪列升序排列的结果集。
<i>fts_timestamp</i> desc	按时间戳降序排列的结果集。
<i>fts_timestamp</i> asc	按时间戳升序排列的结果集。
<i>column_name</i> desc	按某个列的降序排序的结果集。 <i>column_name</i> 是源表的列名。
<i>column_name</i> asc	按某个列的升序排序的结果集。 <i>column_name</i> 是源表的列名。
<i>fts_cluster</i> asc	集群结果集。请参见第 45 页的 “使用伪列请求集群结果集”。

注释 在按特定列排序之前，必须修改 *style.vgw* 和 *style.ufl* 文件。请参见第 28 页的 “设置用作排序规范的列”。

例如，下面的查询将按时间戳的升序对文档排序：

```
select t1.score, t2.copy
```

```

from i_blurbs t1, blurbs t2
where t1.id=t2.id and t1.score > 90
and t1.index_any = "<accrue>(raconteur, Paris)"
and t1.sort_by = "fts_timestamp asc"

```

使用 *summary* 列摘要文档

使用 *summary* 伪列可让查询只返回符合搜索条件的文档摘要，而不返回整个文档。缺省情况下不能使用 *summary* 列；必须在创建启用摘要的文本索引之前编辑 *style.prm* 文件。请参见第 25 页的“启用按示例查询、摘要和集群”。

例如，下面的查询只返回包含有单词“Iranian”和“book”的文档摘要（在本例中，将 *style.prm* 文件配置为显示 255 个字符）：

```

select t1.score, t1.summary
from i_blurbs t1, blurbs t2
where t1.id=t2.id and t1.score > 70
and t1.index_any = "(Iranian <and> book)"

score      summary
-----
78         They asked me to write about myself and my book, so here goes:
          I started a restaurant called "de Gustibus" with two of my fri

(1 row affected)

```

增强型全文本搜索引擎最多支持 255 字节的摘要。

有关使用摘要的查询的其它示例，请参见 *\$SYBASE/\$SYBASE_FTS/sample/scripts* 目录中的脚本示例 *sample_text_queries.sql*。

使用伪列请求集群结果集

集群功能可以分析结果集，并将行分组放入集群中，以便使每个集群中的行彼此之间的语义比与其它集群中的行之间的语义更接近。按子主题对行排序有助于了解结果集所涵盖的主要科目范围。

返回集群结果集可能比返回非集群结果集慢得多。如果查询的响应时间很关键，则使用非集群结果集。

准备使用集群

在请求集群结果集之前，必须建立启用集群功能的文本索引。请参见第 25 页的“启用按示例查询、摘要和集群”。

Verity 集群算法尝试基于下列配置参数的值，将类似的行组合在一起：

- `cluster_style`
- `cluster_max`
- `cluster_effort`
- `cluster_order`

使用 `sp_text_cluster` 让查询使用与这些配置参数的缺省值不同的值。请参见第 137 页的“`sp_text_cluster`”。

编写请求集群结果集的查询

若要获得集群结果集，请在查询的 `sort_by` 伪列中指定“`fts_cluster`”作为排序规范。例如：

```
select t1.score, t2.copy
from i_blurbs t1, blurbs t2
where t1.id=t2.id
and t1.index_any = "<many> <word> software"
and t1.max_docs = 10
and t1.sort_by = "fts_cluster asc"
```

若要返回其它的集群信息，请在查询中包括下列任意伪列：

- `cluster_number` — 包含行所属的集群的编号。集群编号从 1 开始。
- `cluster_keywords` — 包含集群中最常见的词。`cluster_keywords` 列包含多个空值，每个空值对应于一个不符合任何集群的行。
- `max_docs` — 限制为每个集群返回的行数。在非集群查询中，`max_docs` 列限制在一个结果集中返回的总行数。
- `score` — 包含从 0 到 10000 之间的一个值。此得分越高，行就越靠近集群中心。得分为 0 表示行不符合任何集群。在非集群查询中，`score` 列可以是 0 到 100 之间的一个值。搜索引擎不返回得分为 0 的结果。得分为 0 表示“不匹配”，但用户绝对不会看到得分为 0 的情况。

有关使用集群的 SQL 语句的示例，请参见 `$$SYBASE/$SYBASE_FTS/sample/scripts` 目录中名为 `sample_text_queries.sql` 的脚本示例。

全文本搜索运算符

用来执行全文本搜索的特殊搜索运算符是 Verity 搜索引擎的一部分。
表 5-3 介绍增强型全文本搜索引擎提供的 Verity 搜索运算符。

注释 Sybase 不支持将 XML 元素的索引编制到字段中。

表 5-3: Verity 搜索运算符

运算符名称	说明
accrue	选择至少包含查询中指定的一个搜索元素的文档。包含的搜索元素越多，得分就越高。
and	选择包含查询中指定的全部搜索元素的文档。
complement	返回 score 值的补数（用 100 减去 score 值）。
in	选择在指定的文档区域内包含有搜索条件的文档。
like	选择与示例文档或查询中指定的段落相似的文档。
near	选择包含指定搜索元素的文档，搜索术语在文档中彼此越接近，文档的得分就越高。
near/n	选择两个或多个搜索术语相隔在 n 个单词之内的文档，其中 n 是不超过 1000 的整数。搜索术语在文档中彼此越接近，文档的得分就越高。
or	选择至少包含查询中指定的一个搜索元素的文档。
paragraph	选择在同一个段落内包含所有指定搜索元素的文档。
phrase	选择包含特殊短语的文档。短语是以特定顺序出现的两个或更多个单词的组合。
product	将搜索条件的每一项的得分值相乘。
sentence	选择在同一句子中包含所有指定单词的文档。
stem	扩展搜索，使其包含指定的单词及其变化形式。
sum	将搜索条件中每个项目的得分值相加。
thesaurus	扩展搜索，使其包含指定的单词及其同义词。
topic	指定输入的搜索词条是一个主题。
typo/n	扩展搜索，使其包含指定的单词及与其相似的单词。可选的 n 变量指定查询术语与匹配术语错误的最大数目。
wildcard	匹配包括在搜索字符串中的通配符。某些字符自动指示通配符规范。
word	执行一项基本的单词搜索，选择包含指定单词的一个或多个实例的文档。
yesno	将所有非零得分值转换为 100。

使用 Verity 运算符时的注意事项

编写全文本搜索查询时：

- Verity 将破折号、下划线和与符号视为字母字符，而不是字分隔符。可以通过修改特定于区域设置的定义文件来修改此行为。例如：可以将 `-tokenized_as_alphabet-&` 更改为 `-tokenized_as_alphabet_&`。
- 在查询中 *必须* 用尖括号 (`<>`) 将运算符括起来。如果不将它们用尖括号括起来，则增强型全文本搜索引擎将发出如下所示的错误消息：

```
Msg 20200, Level 15, State 0:
Server 'KRAZYKAT', Line 1:
Error    E1-0111 (Query Builder): Syntax error in query string near
character 5
Msg 20200, Level 15, State 0:
Server 'KRAZYKAT', Line 1:
Error    E1-0114 (Query Builder): Error parsing query: word(tasmanian)
Msg 20101, Level 15, State 0:
Server 'KRAZYKAT', Line 1:
VdkSearchNew failed with vdk error (-40).
Msg 20101, Level 15, State 0:
Server 'KRAZYKAT', Line 1:
VdkSearchGetInfo failed with vdk error (-11).
score      copy
-----
(0 rows affected) score
```

- 必须将 Verity 语言查询包括到单引号 (') 或双引号 (") 中。增强型全文本搜索引擎在将查询发送到 Verity 之前会剥离掉最外层的引号。例如，如果输入查询：

```
...where index_any = "'?own'"
```

增强型全文本搜索引擎将向 Verity 发送以下查询：

```
'?own'
```

- 查询可能会由 SQL 中用 `and` 命令连接在一起的几个 “`index_any`” 子句组成。 `and` 命令右边的值字符串可以用 “`<snnn>`” 作为前缀。在增强型全文本搜索中，所有这样的字符串会按 “`nnn`” 值所确定的顺序连接在一起，并删除 “`<snnn>`”。例如：

```
where index_any="<s001>hello"
and index_any="<s002> world"
```

等同于：

```
where index_any = "hello world"
```

对于大于 255 个字符的搜索字符串，这是一个解决方法。

- 以混合大小写输入搜索词条自动具备区分大小写的特性。以全部大写或全部小写输入的搜索词条不会自动具备区分大小写的特性。例如，查询“Server”只能找到“Server”；查询“server”或“SERVER”则能找到“Server”、“server”和“SERVER”。
- 对查询表达式可以使用表 5-4 中所示的替代语法。

表 5-4: Verity 的替代语法

标准查询表达式	替代语法
<MANY><WORD>string	"string"
<MANY><STEM>string	'string'

在使用替代语法时，需要记住一点：增强型全文本搜索引擎将查询发送到 Verity 之前，会剥离掉最外层的引号。例如，如果输入查询：

```
...where index_any = "'play'"
```

增强型全文本搜索引擎将向 Verity 发送以下查询：

```
'play'
```

它等同于：

```
<MANY><STEM>play
```

使用 Verity 运算符

接下来的部分介绍如何使用第 47 页的表 5-3 中所示的 Verity 运算符。

accrue

accrue 运算符选择至少包含查询中所指定的一个搜索项的文档。必须有两个或更多个搜索元素。每个结果都是按相关性排列的。例如，以下查询在 blurbs 表的 copy 列中搜索单词“restaurant”或“deli”，或同时搜索这两个单词：

```
select t1.score, t2.copy
from i_blurbs t1, blurbs t2
where t1.id=t2.id and t1.score > 35
and t1.index_any = "<accrue>(restaurant, deli)"
```

and 和 or

and 和 **or** 运算符选择包含指定的搜索元素的文档。每个结果都是按相关性排列的。**and** 运算符选择包含查询中指定的所有元素的文档。例如，下面的查询选择既包含 “Iranian” 又包含 “business” 的文档：

```
select t2.copy
from i_blurbs t1, blurbs t2
where t1.id=t2.id
and t1.index_any = "(Iranian <and> business)"
```

or 运算符选择包含任意搜索元素的文档。例如，如果重写前面的查询，使用 **or** 运算符，此查询将选择包含单词 “Iranian” 或 “business” 的文档：

```
select t2.copy
from i_blurbs t1, blurbs t2
where t1.id=t2.id
and t1.index_any = "(Iranian <or> business)"
```

complement

complement 运算符返回文档的 **score** 值的补数，也就是，用 100 减去 **score** 值，并返回该结果作为文档的 **score** 值。

in

in 运算符选择在一个或多个文档区域中包含指定的搜索元素的文档。在下面两种情况下将为文本索引创建文档区域：

- 当使用 `sp_create_text_index` 对两个或多个列创建索引时，将为文本索引中的每一列创建一个文档区域。请参见第 19 页的“[创建文本索引时指定多列](#)”。对单个列创建文本索引时不会创建文档区域。例如，如果在创建文本索引时指定 `blurbs` 表中的 `au_id` 和 `copy` 列，可以发出以下查询：

```
select t1.score, t2.copy
from i_blurbs t1, blurbs t2
where t1.id=t2.id and t1.score > 35
and t1.index_any = "gorilla <in> copy"
```

这将返回 `copy` 列中包含单词 “gorilla” 的各行。不过，如果在创建文本索引时仅指定 `blurbs` 表中的 `copy` 列，该查询将不会返回任何行。

- 当创建使用过滤器的索引时，将为文档中的每个标记创建一个文档区域。请参见第 30 页的“对包含标记的文本使用过滤器”。通过在 `in` 运算符后指定标记名称，可以将搜索限定到特定的标记。例如，要在 HTML 文档的“`title`”标记中查找单词“`automotive`”，应指定：

```
select t1.score, t2.copy
from i_blurbs t1, blurbs t2
where t1.id=t2.id and t1.score > 35
and t1.index_any = "automotive <in> title"
```

使用过滤器的文本索引可以只包含一列。

like

`like` 运算符选择与所提供的文档或段落相似的文档。搜索引擎将分析文本，以找到要使用的最重要的词条。如果指定了多个样本，搜索引擎将选择所有样本中公用的重要词条。每个结果都是按相关性排列的。

`like` 运算符接受单个操作数，称为按示例查询 (QBE) 规范。QBE 规范可以是字符文本也可以是文档 ID。文档 ID 来自于源表中的 `IDENTITY` 列。例如，若要选择与 `IDENTITY` 为“2”的行中 `copy` 列内的文档类似的文档，请输入：

```
select t1.score, t2.copy
from i_blurbs t1, blurbs t2
where t1.id=t2.id and t1.score > 35
and t1.index_any = '<like> ( "{2}" )'
```

使用 QBE 规范中的字符文本之前，必须对 `style.prm` 文件中的以下行撤消注释：

```
$define DOC-FEATURES "TF"
```

请参见第 25 页的“启用按示例查询、摘要和集群”。

有关使用 QBE 的 SQL 语句的示例，请参见 `$$SYBASE/$SYBASE_FTS/sample/scripts` 目录中名为 `sample_text_queries.sql` 的脚本示例。

near 和 near/n

near 运算符选择包含查询中指定的项目并且这些项目彼此相隔很近的文档（“近”是一个相对的概念）。搜索单词在文档中相隔越近，文档的相关性排列程度也就越高。

near/n 运算符指定两个项目可以相隔多远（*n* 的最大值为 1000）。下面的示例所选择的文档中，“**raconteur**”和“**home**”单词相距在 10 个单词之内：

```
select t2.copy
from i_blurbs t1, blurbs t2
where t1.id=t2.id
and t1.index_any = "<near/10>(raconteur, home)"
```

or

请参见第 50 页的“**and 和 or**”。

phrase

phrase 运算符选择包含特定短语（一组按特定顺序出现的两个或多个项目）的文档。每个结果都是按相关性排列的。下面的示例将选择包含短语“**gorilla's head**”的文档：

```
select t1.score, t2.copy
from i_blurbs t1, blurbs t2
where t1.id=t2.id and t1.score > 50
and t1.index_any = "<phrase>(gorilla's head)"
```

paragraph

paragraph 运算符选择在同一段落中出现指定的搜索元素的文档。这些单词在段落中相隔越近，文档在相关性排列方面的得分就越高。下面的示例搜索单词“**text**”和“**search**”出现在同一段落内的文档：

```
select t1.score, t2.copy
from i_blurbs t1, blurbs t2
where t1.id=t2.id and t1.score > 50
and t1.index_any = "<many><paragraph>(text, search)"
```

product

product 运算符将每个搜索元素的文档 **score** 值相乘。要得到文档的得分，增强型全文本搜索引擎将计算每个搜索元素的得分，并将各个 **score** 相乘。例如：

```
select t1.score, t2.copy
from i_blurbs t1, blurbs t2
where t1.id=t2.id and t1.score > 50
and t1.index_any = "<product>(cat, created)"
```

每个搜索元素的 **score** 值是 78；然而，因为要将各个项目的 **score** 值相乘，所以文档的 **score** 值是 61 ($.78 \times .78 = .61(100) = 61$)。

sentence

sentence 运算符选择在同一个句子中出现指定的搜索元素的文档。这些单词在句子中相隔越近，文档在相关性排列方面的得分就越高。下面的示例搜索单词 “tax” 和 “service” 出现在同一句子内的文档：

```
select t1.score, t2.copy
from i_blurbs t1, blurbs t2
where t1.id=t2.id and t1.score > 50
and t1.index_any = "<many><sentence>(tax, service)"
```

stem

stem 运算符搜索包含指定单词及其变化形式的文档。例如，如果指定单词 “cook”，增强型全文本搜索引擎将找到包含 “cooked”、“cooking”、“cooks” 等的文档。若要对结果集进行相关性排列，请在查询中包括 **many** 修饰符。请参见第 57 页的 “运算符修饰符”。

下面的查询使用 **stem** 运算符查找包含单词 “create” 的各种变化形式的文档，其变化形式也就是以 “create” 作为词根的单词。注意，即使第一个文档所包含的单词不以 “create” 作为完整的词根（如 “creative”），但仍然会选择该文档：

```
select t1.score, t2.copy
from i_blurbs t1, blurbs t2
where t1.id=t2.id and t1.score > 10
and t1.index_any = "<many><stem>create"
score      copy
-----
78         Anne Ringer ran away from the circus as a child. A
          university creative writing professor and her family
. . .
78         If Chastity Locksley didn't exist, this troubled world
          would have created her! Not only did she master the mystic
```

sum

sum 运算符合计每个搜索元素的 **score** 值，直至最大值 100。为得到文档的得分，增强型全文本搜索引擎计算每个搜索元素的得分，并累加这些得分。

thesaurus

thesaurus 运算符搜索包含搜索元素同义词的文档。例如，您可以使用单词 “dog” 来查询包含其同义词（如 “canine”、“pooch”、“pup”、“watchdog” 等）的文档。每个结果都是按相关性排列的。

增强型全文本搜索引擎提供缺省的同义词库。您还可以创建自定义的同义词库。请参见第 32 页的 “创建自定义同义词库”。

下面的示例使用 **thesaurus** 运算符来查找包含单词 “crave” 的同义词的结果集。第一个文档被选中，因为它包含单词 “want”；第二个文档也被选中，因为它包含单词 “hunger”：

```
select t2.copy
from i_blurbs t1, blurbs t2
where t1.id=t2.id
and t1.index_any = "<thesaurus>(crave)"

score      copy
-----
78      They asked me to write about myself and my book, so here
        goes:  I started a restaurant called "de Gustibus" with two
        . . .
        of restaurant over another, when what they really want is a
        . . .
78      A chef's chef and a raconteur's raconteur, Reginald
        Blotchet-Halls calls London his second home. "Th' palace
        . . .
        his equal skill in satisfying our perpetual hunger for
        . . .
```


topic

`topic` 运算符选择符合指定主题所定义搜索条件的文档。请参见第 35 页的“创建主题”。例如，使用下面语法查找符合由主题“engineering”所定义搜索条件的文档：

```
select t2.copy
from i_blurbs t1, blurbs t2
where t1.id=t2.id
and t1.index_any = "<topic>(engineering)"
```

typo/n

`typo/n` 运算符将搜索扩展为包括指定单词及其类似单词。可选的 n 变量指定查询术语与匹配术语错误的最大数目。例如，`typo mouse` 返回包含如“house”、“louse”和“moose”等单词的文档。

wildcard

`wildcard` 运算符允许使用通配符替代所搜索项目的一部分。表 5-5 介绍通配符及其属性。

表 5-5：增强型全文本搜索引擎的通配符

字符	作用	语法	查找
?	指定一个字母数字字符。当查询中包括问号时，不必包括 <code>wildcard</code> 运算符。在集合 ([]) 或可选模式 ({}) 中，问号将被忽略。	'?an'	“ran”、“pan”、“can”和“ban”
*	指定零个或多个字母数字字符。当在查询中包括星号时不必包括 <code>wildcard</code> 运算符；不能使用星号指定通配符字符串的第一个字符。在集合 ([]) 或可选模式 ({}) 中，星号将被忽略。	'corp*'	“corporate”、“corporation”、“corporal”和“corpulent”
[]	指定集合中的任意单个字符。如果单词中包括集合，则必须将单词用反引号 (` `) 括起来。另外，集合中不能有空格。	<wildcard> 'c[auo]t'	“cat”、“cut”和“cot”
{ }	指定其中一个模式，各模式间用逗号分开。如果单词中包括模式，则必须将单词用反引号 (` `) 括起来。另外，集合中不能有空格。	<wildcard> 'bank{s,e,r,ing}'	“banks”、“banker”和“banking”
^	指定任意一个未包括在集合中的字符。尖号 (^) 必须是引入集合的左括号 ([) 后的第一个字符。	<wildcard> 'st[^oa]ck'	查找“stick”和“stuck”，但不包括“stock”和“stack”
-	指定集合中字符的范围。	<wildcard> 'c[a-r]t'	包括从“cat”到“crt”的所有三个字母单词

若要对结果集进行相关性排列，请在查询中包括 `many` 修饰符。请参见第 57 页的“运算符修饰符”。

例如，下面的查询将搜索包括单词 “slingshot” 变化形式的文档：

```
select t2.copy
from i_blurbs t1, blurbs t2
where t1.id=t2.id
and t1.index_any = 'slingshot*'
score      copy
-----
100      Albert Ringer was born in a trunk to circus parents, but
         another kind of circus trunk played a more important role
         . . .
         gorilla. "Slingshotting" himself from the ring ropes,
         . . .
```

word

`word` 运算符搜索包含指定单词的文档。若要对结果集进行相关性排列，请将 `many` 运算符包括在查询中。下面的示例将搜索 `blurbs` 表以查找包含单词 “palates” 的文档：

```
select t1.score, t2.copy
from i_blurbs t1, blurbs t2
where t1.id=t2.id and t1.score > 50
and t1.index_any = "<many><word>(palates)"
```

yesno

`yesno` 运算符将所有非零的 `score` 值转换为 100。例如，如果五个文档的得分分别是 86、45、89、89 和 100，那么随着其中每个文档返回的 `score` 值均为 100。为 0 的 `score` 值不更改。`yesno` 运算符有助于保证所有包含搜索条件的文档都返回到结果集中，而不管其排序顺序如何。

运算符修饰符

Verity 查询语言包括一些修饰符，这些修饰符可与运算符一起使用以优化搜索。表 5-6 介绍了这些修饰符。

表 5-6: Verity 运算符的修饰符

修饰符名称	说明	与以下运算符一起使用	示例
case	执行区分大小写的搜索。如果输入的搜索词条大小写混合，搜索将自动区分大小写。	wildcard word	<case><word> (Net)
many	统计单词、带词根的单词或短语在文档中出现的次数。根据其密度对文档进行相关性排列。	paragraph phrase sentence stem word wildcard	<many><stem> (write)
not	排除包含查询所搜索的项目的文档。	and or	cat<and><not>elephant
order	指定文档中的项目按其在查询中出现的顺序出现。始终将 order 修饰符放在紧靠运算符的前面	near/n paragraph sentence	简单语法： tidbits<order><paragraph>king 更清楚的语法： <order><paragraph> (tidbits, king)

本章介绍增强型全文本搜索引擎的系统管理问题。

主题	页码
在 UNIX 上启动增强型全文本搜索引擎	59
在 Windows NT 上启动增强型全文本搜索引擎	60
关闭增强型全文本搜索引擎	62
修改配置参数	62
增强型全文本搜索引擎的备份和恢复	70

在 UNIX 上启动增强型全文本搜索引擎

使用 `startserver` 实用程序可在 UNIX 上启动增强型全文本搜索引擎。`startserver` 实用程序位于 Adaptive Server 的 `install` 目录下。例如，若要启动名为 KRAZYKAT 的增强型全文本搜索引擎，请输入：

```
startserver -f
$SYBASE/$SYBASE_FTS/install/RUN_KRAZYKAT
```

其中，`-f` 标志指定 `runserver` 文件的相对路径。发布该命令后，增强型全文本搜索引擎将发布一系列消息，描述配置参数的设置。

创建 runserver 文件

`runserver` 文件包含增强型全文本搜索引擎的启动命令。`runserver` 文件可以包含以下标志：

- `-Sserver_name` — 指定增强型全文本搜索引擎的名称，用于查找配置文件以及 `interfaces` 文件中的网络连接信息。
- `-t` — 让增强型全文本搜索引擎将启动消息写成标准错误。
- `-lerrorlog_path` — 指定错误日志文件的路径。
- `-iinterfaces_file_path` — 指定 `interfaces` 文件的路径。

在安装期间，一个示例 `runserver` 文件被复制到 `$$SYBASE/$SYBASE_FTS/install` 目录中。创建该文件的一个副本，将它重命名为 `RUN_server_name`，其中，`server_name` 是增强型全文本搜索引擎的名称。必须在 `runserver` 文件中包含所用平台的正确路径环境变量。表 6-1 显示了每个平台所用的路径环境变量。

表 6-1: runserver 文件的路径环境变量

平台	环境变量
RS/6000 AIX	LIBPATH
Sun Solaris	LD_LIBRARY_PATH
HP 9000(800)	SHLIB_PATH
Compaq Tru64	LD_LIBRARY_PATH
Linux	LD_LIBRARY_PATH
NT	PATH

例如，Sun Solaris 上名为 `KRAZYKAT` 的增强型全文本搜索引擎的 `runserver` 文件可能是 `RUN_KRAZYKAT`，并且该文件类似于：

```
#!/bin/sh
#

LD_LIBRARY_PATH="$SYBASE/$SYBASE_FTS/lib:$LD_LIBRARY_PATH"
export LD_LIBRARY_PATH

$SYBASE/bin/txtsvr -SKRAZYKAT
```

`runserver` 文件中的启动命令必须由一行构成，并且不能包含回车符。如果文件的内容必须使用两行或三行，则应在每个换行处使用一个反斜杠 (`\`)。

在 Windows NT 上启动增强型全文本搜索引擎

可从 Sybase Central™ 启动增强型全文本搜索引擎作为服务，或从以下命令行启动：

- 从 Sybase Central — 有关启动服务器的信息，请参见 Sybase Central 文档。
- 作为服务 — 请参见第 61 页的“启动增强型全文本搜索引擎作为服务”

- 从命令行 — 使用下面的语法：

```
%SYBASE%\%SYBASE_FTS%\bin\txtsvr.exe -Sserver_name
[-t] [-i%SYBASE%path_to_sql.ini_file] [-l%SYBASE%path_to_errorlog]
```

其中：

- -S 是所启动的增强型全文本搜索引擎的名称。
- -t 将启动消息指示为标准错误。
- -i 是 *sql.ini* 文件的路径。此路径必须包含 *sql.ini* 文件名。
- -l 是错误日志的路径。此路径必须包含错误日志文件的名称。

例如，若要使用缺省的 *sql.ini* 和错误日志文件在 NT 上启动名为 KRAZYKAT 的增强型全文本搜索引擎，并使用 -t 跟踪启动消息，请输入：

```
%SYBASE%\%SYBASE_FTS%\bin\txtsvr.exe -SKRAZYKAT -t
```

当看到启动完成消息时，即说明增强型全文本搜索引擎已在运行。

启动增强型全文本搜索引擎作为服务

使用 Sybase Central 中的 *instsvr* 实用程序可将增强型全文本搜索引擎添加到项目列表，该添加操作可用 *Services* 实用程序开始或停止。*instsvr* 位于 *%SYBASE%\%SYBASE_FTS%\bin* 目录中。

instsvr 实用程序使用以下语法：

```
instsvr.exe service_name %SYBASE%\%SYBASE_FTS%\bin\txtsvr.exe
"startup_parameters"
```

其中：

- *service_name* — 是要作为服务添加的增强型全文本搜索引擎的名称。对于 Sybase Central，Sybase 建议使用带有扩展 “_TS” 的服务名（例如，KRAZYKAT_TS）。
- *startup_parameters* — 是启动时要使用的任何参数。

例如，若要将名为 KRAZYKAT_TS 的增强型全文本搜索引擎作为服务安装，请输入：

```
instsvr.exe KRAZYKAT_TS %SYBASE%\sds\text\bin\txtsvr.exe
"-SKRAZYKAT_TS -t"
```

注释 若要包含多个参数（例如，`-i`），请用一对双引号将所有参数括起来。

若要将 Sybase Central 配置为启动和停止增强型全文本搜索引擎，必须提供一个以 “SYBTEXT_*server_name*” 开头的服务名，其中，*server_name* 是 `interfaces` 文件中所列增强型全文本搜索引擎的名称。例如，如果 `interfaces` 文件中的名称为 `KRAZYKAT_TS`，则运行以下 `instsvr` 命令将创建一个可由 Sybase Central 管理的服务：

```
instsvr SYBTEXT_KRAZYKAT_TS %SYBASE%\%SYBASE_FTS%\bin\txtsvr.exe  
"-SKRAZYKAT_TS -t"
```

关闭增强型全文本搜索引擎

使用下面的命令可从 Adaptive Server 关闭增强型全文本搜索引擎：

```
server_name...sp_shutdown
```

其中，*server_name* 是所关闭的增强型全文本搜索引擎的名称。

只有具有 `sa_` 角色的用户才能关闭增强型全文本搜索引擎。

例如，若要关闭名为 `KRAZYKAT` 的增强型全文本搜索引擎，请输入：

```
KRAZYKAT...sp_shutdown
```

修改配置参数

每个增强型全文本搜索引擎都提供一些具有缺省值的配置参数，如表 6-2 所示。

表 6-2: 配置参数

参数	说明	缺省值
<code>batch_size</code>	确定发送到增强型全文本搜索引擎的批处理大小。	500
<code>batch_blocksize</code>	启用时，文本服务器以更小块的方式读取数据。该参数指示文本服务器一次检索 <i>n</i> 行数据。参数值介于 0（表示禁用）和 65535 之间。	0

参数	说明	缺省值
max_indexes	增强型全文本搜索引擎中创建的最大文本索引数。	126
max_stacksize	为客户端线程分配的堆栈大小（以 KB 表示）。	34,816
max_threads	增强型全文本搜索引擎中可用的最大线程数。	50
max_packetsize	在增强型全文本搜索引擎与 Adaptive Server 之间发送的包大小。	2048
max_sessions	增强型全文本搜索引擎的最大会话数。	100
min_sessions	增强型全文本搜索引擎的最小会话数。	10
language	增强型全文本搜索引擎使用的语言。	us_english
charset	增强型全文本搜索引擎使用的字符集。	iso_1
vdkCharset	Verity 搜索引擎使用的字符集。	850
vdkLanguage	Verity 搜索引擎使用的语言。	english
vdkHome	Verity 目录。	UNIX: \$SYBASE/\$SYBASE_FTS/verity Windows NT: %SYBASE%\%SYBASE_FTS%\verity
collDir	增强型全文本搜索引擎的集合的存储位置。	UNIX: \$SYBASE/\$SYBASE_FTS/collections Windows NT: %SYBASE%\%SYBASE_FTS%\collections
defaultDb	存储文本索引元数据的增强型全文本搜索引擎数据库的名称。	text_db
interfaces	增强型全文本搜索引擎使用的 interfaces 文件所在目录的完整路径。	UNIX: \$SYBASE/interfaces Windows NT: %SYBASE%\ini\sql.ini
sort_order	缺省排序顺序。	0
errorLog	错误日志文件的完整路径名。	从中启动增强型全文本搜索引擎的目录
traceflags	包含用于生成诊断信息的数字标识符的字符串。	0
srv_traceflags	包含数字标识符的字符串，这些数字标识符用于生成 Open Server 诊断信息。	0
max_session_fd	增强型全文本搜索会话所用的文件描述符的最大数目。请参见第 79 页的“文件描述符和增强型全文本搜索”。	0
cluster_style	使用的集群风格。	Fixed

参数	说明	缺省值
cluster_max	当 cluster_style 设置为 Fixed 时生成的集群的最大数目。	0
cluster_effort	增强型全文本搜索引擎查找一个合适的集群应花费的时间。	缺省值
cluster_order	返回集群以及集群内行的顺序。	0
auto_online	指定启动增强型全文本搜索引擎时，是否使索引自动联机。0 表示不自动联机；1 表示自动联机。	0
backDir	放置文本索引备份文件的缺省位置。	UNIX: \$SYBASE/\$SYBASE_FTS/backup Windows NT: %SYBASE%\%SYBASE_FTS%\backup
knowledge_base	用于实现 Verity 主题功能的知识库映射的位置。	空
nocase	设置增强型全文本搜索引擎是否区分大小写。如果在 Adaptive Server 中使用区分大小写的排序顺序，该值设置为 0。如果使用不区分大小写的排序顺序，则设置为 1。	0

安装期间，会将一个包含所有这些参数的示例配置文件复制到安装目录下。附录 B “示例文件” 中提供了一个名为 *textsvr.cfg* 的示例配置文件。

修改配置值

使用 `sp_text_configure` 可更改配置参数的值。语法为：

```
sp_text_configure server_name, config_name, config_value
```

其中：

- *server_name* — 是增强型全文本搜索引擎的名称。
- *config_name* — 是配置参数的名称。
- *config_value* — 是赋给配置参数的值。

请参见第 139 页的 “`sp_text_configure`”。

可用的配置参数

表 6-3 提供了可用配置参数的列表，这些参数带有有效限制：

表 6-3：配置参数的限制

参数	值	静态 / 动态
batch_size	0 — MAX_INT	动态
batch_blocksize	0 — 65535	动态
max_indexes	0 — MAX_INT	静态
max_stacksize	0 — MAX_INT	静态
max_threads	0 — MAX_INT	静态
max_packet_size	0 — MAX_INT	静态
max_sessions	0 — MAX_INT	静态
min_sessions	0 — max_sessions	静态
language	french、spanish、german、us_english	静态
charset	ascii_8、cp037、cp1047、cp437、cp500、cp850、deckanji、eucjis、iso_1、mac、roman8、sjis、utf8	静态
vdkCharset	50、437、1252、mac1（即手册中所列的值）	静态
vdkLanguage	frenchx、spanishx、germanx、english、englishx、bokmalx、dutchx、finnishx、nynorskx、swedishx、portugx、italianx、danishx	静态
vdkHome	一个字符数 < 255 的字符串	静态
collDir	一个字符数 < 255 的字符串	静态
default_Db	一个字符数 < 32 的字符串	静态
interfaces	一个字符数 < 255 的字符串	静态
sort_order	0, 1, 2, 3	动态
errorLog	一个字符数 < 255 的字符串	静态
traceflags	一个带有用逗号分隔的数字（范围从 1 到 15）的字符串。	静态
srv_traceflags	一个带有用逗号分隔的数字（范围从 1 到 8）的字符串。	静态
cluster_style	Coarse、Medium、Fine、Fixed	动态
cluster_max	0 — MAX_INT	动态
cluster_effort	Low、Medium、High、Default	动态
cluster_order	0 或 1	动态
auto_online	0 或 1	静态
backCmd	一个字符数 < 255 的字符串	动态
restoreCmd	一个字符数 < 255 的字符串	动态
backDir	一个字符数 < 255 的字符串	静态
knowledge_base	一个字符数 < 255 的字符串	静态
nocase	0 或 1	动态
max_session_fd	0, 5 — MAX_INT	静态

设置缺省语言

Verity 的缺省语言是用 `vdkLanguage` 配置参数设置的。缺省情况下，`vdkLanguage` 会设置为 “english”。可以配置 Verity 使用另外一种缺省语言。表 6-4 列出了 Sybase 支持的区域设置。

表 6-4: vdkLanguage 配置参数

语言	缺省区域设置名
英语	english
德语	german
法语	french

其它语言配套程序可在 `$$SYBASE/$$SYBASE_FTS/verity/common` 目录中获得；但是，增强型全文本搜索引擎仅以表 6-4 中所示的语言显示消息。

`language` 参数是增强型全文本搜索引擎用来显示其错误消息以及 Open Server 和 Open Client 错误消息的语言。将 `language` 参数设置为 Adaptive Server 语言。

使用增强型全文本搜索引擎，运行：

```
sp_text_configure KRAZYKAT, 'vdkLanguage', 'spanish'
```

有关 Verity 语言的详细信息，请参见位于 <http://www.verity.com> 的 Verity 网站。

设置缺省字符集

可使用配置文件中的 `vdkCharset` 参数设置 Verity 的缺省字符集。Verity 字符集所用的这些文件位于 `$$SYBASE/$$SYBASE_FTS/verity/common` 中。表 6-5 描述了可以用于 Verity 的字符集。

表 6-5: Verity 字符集

字符集	说明
850	缺省值
437	IBM PC 字符集
1252	用于西欧语言的 Windows 代码页
mac1	Macintosh 罗马字符集

可以用 `charset` 参数设置增强型全文本搜索引擎的缺省字符集。将 `charset` 参数设置为 Adaptive Server 字符集。

例如，使用增强型全文本搜索引擎，运行：

```
sp_text_configure KRAZYKAT, 'vdkCharset', '437'
```

对欧元符号编制索引

若要对欧元符号编制索引，请在 Adaptive Server 中安装 utf8 字符集。将 vdkLanguage 设置为 <语言>x，将 vdkCharset 保留为空白。例如：

```
ASE 12.5.x charset           =   utf8
EFTS 12.5.x vdkLanguage     =   englishx
EFTS 12.5.x vdkCharset      =
```

设置缺省排序顺序

缺省情况下，增强型全文本搜索引擎按 score 伪列的降序（高分显示在前）对结果集进行排序。若要改变缺省的排序顺序，请将 sort_order 配置参数设置为表 6-6 中所列的值之一。

表 6-6：配置文件的排序顺序值

值	说明
0	返回按 score 伪列降序排列的结果集。缺省值。
1	返回按 score 伪列升序排列的结果集。
2	返回按时间戳降序排列的结果集。
3	返回按时间戳升序排列的结果集。

例如，输入：

```
sp_text_configure KRAZYKAT, 'sort_order', '2'
```

当按时间戳降序对结果集排序（表 6-6 中的值 2）时，增强型全文本搜索引擎将首先返回最新的文档。最新文档是那些最近插入或更新的文档。当按时间戳升序对结果进行排序（表 6-6 中的值 3）时，增强型全文本搜索引擎将首先返回最早的文档。

如果查询使用 max_docs 伪列，则设置缺省排序顺序非常重要。max_docs 伪列将结果集的行数限制为按排序顺序排序后的前 *n* 行。如果将 max_docs 设置为小于结果集大小的一个数，则所选的排序顺序可能不包括带有要搜索信息的行。

例如，如果按时间戳升序排列，则添加到表的最新文档将显示在结果集的最后。如果整个结果集由 11 个文档组成，而将 max_docs 设置为 10，则最新文档不显示在结果集中。但是，如果按时间戳降序排列，则最新文档将显示在结果集的最前面。

设置跟踪标志

当增强型全文本搜索引擎中发生某些事件时，`traceflags` 参数会启用这些事件的记录。每个跟踪标志都用一个数字唯一地进行标识。跟踪标志在表 6-7 中介绍。

表 6-7：增强型全文本搜索引擎的跟踪标志

跟踪标志	说明
1	跟踪 Adaptive Server 中的连接、断开连接和关注事件。
2	跟踪语言事件。跟踪 Adaptive Server 发送到增强型全文本搜索引擎的 SQL 语句。
3	跟踪 RPC 事件。
4	跟踪游标事件。跟踪 Adaptive Server 发送到增强型全文本搜索引擎的 SQL 语句。
5	将显示的错误写入日志。
6	跟踪有关文本索引的信息。将传递到 Verity 的搜索字符串写入日志，并将搜索返回的记录数写入日志。
7	跟踪完成包。
8	跟踪对增强型全文本搜索引擎与 Verity API 之间接口的调用。
9	跟踪 SQL 分析。
10	跟踪 Verity 处理。
11	禁用 Verity 集合优化。
12	禁止 <code>sp_statistics</code> 返回信息。
13	跟踪备份操作。
14	记录 Verity 状态和计时信息。
15	生成集合的 ngram 索引信息。ngram 会提高通配符搜索的速度。对 Unicode 格式的数据进行通配符搜索时需要此跟踪标志。
30	此跟踪标志启用可删除过期集合文件的 Verity MaxClean 功能。该功能仅应当在维护期间使用，因为它将花费额外时间并干扰正常使用。它与 <code>sp_optimize_text_index</code> 一起启用。

可以使用增强型全文本搜索引擎中的远程过程调用 (RPC) `sp_traceon` 和 `sp_traceoff` 交互式启用和禁用跟踪标志。

若要执行 `sp_traceon`，请使用以下命令（其中，`textserver` 是增强型全文本搜索引擎的名称）：

```
textserver...sp_traceon 1,2,3,4
```

跟踪标志将保持活动状态，直到会话终止或直到使用特定的跟踪标志执行 `sp_traceoff` RPC。若要永久地设置一个跟踪标志，既可在配置文件中设置，也可使用 `sp_text_configure` 命令设置。

设置 Open Server 跟踪标志

使用 `srv_traceflags` 参数可启用跟踪标志以记录 Open Server 诊断信息。Open Server 跟踪标志在表 6-8 中介绍。

表 6-8: Open Server 跟踪标志

跟踪标志	说明
1	跟踪 TDS 标头
2	跟踪 TDS 数据
3	跟踪关注事件
4	跟踪消息队列
5	跟踪 TDS 标识
6	跟踪 Open Server 事件
7	跟踪延迟事件队列
8	跟踪网络请求

例如：

```
sp_text_configure KRAZYKAT, 'srv_traceflags', '3'
```

设置区分大小写参数

缺省情况下，增强型全文本搜索引擎是区分大小写的。这意味着必须以相同的大小写输入标识符，否则将无法识别标识符。例如，如果有一个名为 `blurbs`（小写）的表，则不能发布一个指定该表名为 `BLURBS` 的 `sp_create_text_index` 命令。必须发布一个使用相同大小写的表名参数的命令：

```
sp_create_text_index "KRAZYKAT", "i_blurbs", "blurbs", "", "copy"
```

对于增强型全文本搜索引擎，请使用 `nocase` 参数设置增强型全文本搜索引擎是否区分大小写。0 表示区分大小写；1 表示不区分大小写。将 `nocase` 参数设置为 Adaptive Server 中的排序顺序大小写区分性质。

此示例将 KRAZYKAT 服务器更改为不区分大小写。

```
sp_text_configure KRAZYKAT, 'nocase', '1'
```

注释 `nocase` 参数不会影响 Verity 查询的大小写区分性质。请参见第 48 页的“使用 Verity 运算符时的注意事项”。

增强型全文搜索引擎的备份和恢复

Enhanced Full-Text Search Specialty Data Store 的备份和恢复是通过 `sp_text_dump_database` 和 `sp_text_load_index` 自动完成的。这些系统过程为维护数据和文本索引的完整性提供了一个无缝接口。

Adaptive Server 用户数据库和 Verity 集合在物理上是独立的。备份用户数据库不会备份 Verity 集合，且从一个备份恢复数据库不会恢复 Verity 集合。《系统管理指南》第 27 章“备份和恢复用户数据库”中所描述的备份和恢复过程仅适用于 Adaptive Server 中的用户数据库和 `text_db` 数据库。

遵循关于备份数据库的建议计划，如《系统管理指南》第 26 章“制定备份和恢复计划”中所述。当备份带有文本索引的用户数据库时，Sybase 建议同时备份：

- `text_db` 数据库
- 文本索引

注释 不必同时备份用户数据库和文本索引，也可恢复文本索引。但是，在恢复文本索引之前必须先恢复用户数据库。这将恢复 `text_events` 表，`sp_text_load_index` 使用该表可使文本索引与用户数据库同步。

定期安排备份可确保文本索引、Adaptive Server 数据和 `text_events` 表的完整性，有了它们才能在不删除并重建文本索引的情况下恢复这些文本索引。

可自定义的备份和恢复

`backCmd` 和 `restoreCmd` 允许在备份集合文件时，使用可自定义的备份和恢复命令，而不使用 `tar` 或 `zip` 命令。如果 `backCmd` 和 `restoreCmd` 为空，则将使用缺省的命令；否则执行指定的命令。执行命令前，将执行字符串替换以允许指定输入和输出目录以及集合标识符。字符串替换的定义如下：

- `${backDir}` 替换为“backDir”配置参数中指定的 `backup` 目录。
- `${collDir}` 替换为集合的完整路径名。
- `${colID}` 替换为集合 ID，该 ID 是备份文件的全名。

备份 Verity 集合

`sp_text_dump_database` 系统过程将备份集合，也可选择备份用户数据库和 `text_db` 数据库。`sp_text_dump_database` 还会删除恢复时不再需要的项目，以维护 `text_events` 表。

备份期间，增强型全文本搜索引擎处理查询，但会延迟所有的更新请求，直到备份完成。这样便无需关闭再重新启动增强型全文本搜索引擎。

从包含要备份的文本索引的数据库运行 `sp_text_dump_database`。当发布 `sp_text_dump_database` 时，确保所有需要的服务器处于运行状态。`sp_text_dump_database` 无条件地备份所有增强型文本服务器的所有索引。该文本索引的备份存放在由配置参数 `backDir` 指定的目录中。`dump database` 的输出结果写入到增强型全文本搜索的错误日志中。Sybase 建议在备份文本索引时，转储当前数据库和 `text_db` 数据库。不过，这是可选的。

例如，若要备份文本索引，并将 `sample_colors_db` 数据库备份到 `/work2/sybase/colorsbackup` 目录，将 `text_db` 数据库备份到 `/work2/sybase/textdbbackup` 目录，请输入：

```
sp_text_dump_database @backupdbs =  
INDEXES_AND_DATABASES, @current_to = "to  
'/work2/sybase/colorsbackup'", @textdb_to="to  
'/work2/sybase/textdbbackup' "
```

注释 无论何时备份文本索引，备份 `text_db` 数据库都很重要，因为该数据库包含所有文本索引的元数据。

如果所需文件大小大于 2GB，则在 Solaris 上执行 `sp_text_dump_database` 可能会失败。

请参见第 140 页的 `sp_text_dump_database`。

从备份中恢复集合和文本索引

`sp_text_load_index` 系统过程可恢复使用 `sp_text_dump_database` 系统过程备份的文本索引。

数据库管理员可执行以下过程来恢复 Verity 集合：

- 1 恢复 Adaptive Server 用户数据库和 `text_db` 数据库。此过程将源表、元数据和 `text_events` 表恢复到一致的且可预测的状态。遵循《系统管理指南》第 27 章“备份和恢复用户数据库”中所描述的步骤恢复用户数据库和 `text_db` 数据库。
- 2 运行 `sp_text_load_index` 可从最新的索引转储中恢复 Verity 集合。该过程会将最近一次索引转储后产生的所有 `text_events` 表条目的状态重新设置为“unprocessed”，并且会通知增强型全文搜索引擎处理这些事件。

示例：若要恢复 `sample_colors_db` 数据库以及它的所有文本索引：

- 1 恢复 `text_db` 数据库：

```
1> use master
2> go
1> load database text_db from '/work2/sybase/textdbbackup'
2> go
```

- 2 恢复 `sample_colors_db` 数据库：

```
1> load database sample_colors_db from '/work2/sybase/colorsbackup'
2> go
```

- 3 使 `text_db` 和 `sample_colors_db` 数据库联机：

```
1> online database text_db
2> online database sample_colors_db
3> go
```

- 4 恢复文本索引：

```
1> use sample_colors_db
2> go
1> sp_text_load_index
2> go
```

请参见第 144 页的 `sp_text_load_index`。

性能和调优

增强型全文本搜索引擎附带了缺省配置。您可以更改缺省配置，使其更好地反映站点的需要，从而优化增强型全文本搜索引擎的性能。本章介绍可用于提高性能的方法。

主题	页码
更新现有索引	73
提高查询性能	74
重新配置 Adaptive Server	75
重新配置增强型全文本搜索引擎	76
使用 <code>sp_text_notify</code>	77
配置多个增强型全文本搜索引擎	77
多用户	79
文件描述符和增强型全文本搜索	79

更新现有索引

通过启用（打开）跟踪标志 11 和 / 或跟踪标志 12，可减少更新文本索引中的记录所花的时间。

- 启用跟踪标志 11 将禁用 Verity 集合优化。这意味着发出 `sp_text_notify` 后 Verity 不会优化文本索引，从而可以提高性能。如果关闭跟踪标志 11（此为缺省设置），则在 `sp_text_notify` 处理结束时，增强型全文本搜索引擎会调用 Verity 以优化文本索引，这样会推迟 `sp_text_notify` 的完成时间。

安装 Enhanced Full-Text Search Specialty Data Store 之后，如果启用了跟踪标志 11，则可以稍后使用 `sp_optimize_text_index` 来优化文本索引。请参见第 133 页的“[sp_optimize_text_index](#)”。

- 启用跟踪标志 12 将禁止增强型全文本搜索引擎返回 `sp_statistics` 信息。如果关闭了跟踪标志 12（此为缺省设置），则会向增强型全文本搜索引擎发送一个 `update statistics` 命令，该命令会推迟 `sp_text_notify` 的完成时间。

如果文本索引每隔几秒钟就要更新，则对大多数更新而言，都可以通过禁用 `update statistics` 处理和 / 或 Verity 优化来提高性能。

可使用增强型全文本搜索引擎中的 `sp_traceon` 和 `sp_traceoff`，以交互式方式启用和禁用跟踪标志 11 和 12。

提高查询性能

可以通过两种方式显著提高查询性能：

- 限制增强型全文本搜索引擎返回的行数
- 确保查询的连接次序正确

限制行数

使用 `max_docs` 伪列可限制增强型全文本搜索引擎返回的行数。增强型全文本搜索引擎返回的行数越少，Adaptive Server 处理源表与索引表之间的连接的速度就越快。

确保查询的连接次序正确

连接中列出的表和文本索引越多，由于连接次序不对而导致查询速度变慢的可能性就越大。在文本索引与一个或多个表连接时，如果首先查询该文本索引，则查询的运行速度最快。

确保连接次序正确：

- 确保在要索引的表的 `IDENTITY` 列上创建唯一聚簇索引或非聚簇索引
- 将连接限制为一个基表和一个文本索引

如果查询运行很慢，可使用 `showplan` 或启用跟踪标志 11205，并检查连接次序。跟踪标志 11205 会将远程查询转储到 Adaptive Server 错误日志文件。最快的查询是在 `where` 子句中包含 `index_any` 搜索条件，并首先查询文本索引。

最慢的查询是在文本索引的 `where` 子句中包含 `id` 列，并首先查询索引表。在这种情况下，请重写查询或使用 `forceplan` 强制使用查询中列出的连接次序。有关 `forceplan` 的详细信息，请参见 *Performance and Tuning Guide: Optimizer and Abstract Plans*（《性能和调优指南：优化程序和抽象计划》）的第3章“Advanced Optimizing Techniques”（高级优化技术）。

重新配置 Adaptive Server

可通过重新设置本部分讨论的 Adaptive Server 配置参数来提高增强型全文本搜索引擎的性能。（有关用 `sp_configure` 设置配置参数的信息，请参见《系统管理指南》的第4章“设置配置参数”。）

cis cursor rows

`cis cursor rows` 参数指定在一次获取操作期间 Adaptive Server 接收的行数。缺省的 `cis cursor rows` 数为 50。增大该数字将增加在一次获取操作期间 Adaptive Server 从增强型全文本搜索引擎接收的行数。不过，为 `cis cursor rows` 设置的数越大，Adaptive Server 为返回结果集而动态分配的内存就越多。

cis packet size

`cis packet size` 决定了一个网络包中包含的字节数。`cis packet size` 的缺省值为 512。必须为该参数指定 512 的倍数值。增大该参数可提高增强型全文本搜索引擎的性能，原因是包越大，每次查询返回的包就越少。不过，为 `cis packet size` 设置的数越大，Adaptive Server 为该参数分配的内存就越多。

`cis packet size` 参数是动态参数，无需重新启动 Adaptive Server 即可生效。

注释 如果更改 `cis packet size`，则必须同时将增强型全文本搜索引擎配置文件中的 `max_packet_size` 更改为相同的值。如果使用组件集成服务来访问其它远程服务器，则还必须相应地增加那些服务器上的最大网络包大小。

必须重新启动增强型全文本搜索引擎，才能使 `max_packet_size` 参数生效。

重新配置增强型全文本搜索引擎

可通过重新配置本部分讨论的增强型全文本搜索引擎配置参数来提高增强型全文本搜索引擎的性能。请参见第 62 页的“修改配置参数”。

batch_size

batch_size 决定增强型全文本搜索引擎每一批索引的行数。**batch_size** 缺省值为 500（也就是说，每批索引 500 行数据）。增大索引的批大小可提高性能。不过，批大小越大，增强型全文本搜索引擎为该参数分配的内存就越多。

在考虑设置多大的 **batch_size** 时，应考虑要创建文本索引的数据的大小。创建文本索引时，增强型全文本搜索引擎分配的内存等于（以字节为单位）：

(数据所需的空间) x (batch_size) = 所用内存空间

例如，如果索引的数据是每行 10,000 字节，并且 **batch_size** 设置为 500，则当增强型全文本搜索引擎创建文本索引时，需要分配近 5MB 的内存。

可根据数据通常的大小和计算机上的可用内存量来选择批大小。

min_sessions* 和 *max_sessions

min_sessions 和 **max_sessions** 决定了增强型全文本搜索引擎所允许的用户连接数量的最小值和最大值。每个用户连接大约需要 5MB 的内存。不要将 **max_sessions** 设置为超出可用内存的数量。另外，因为用于 **min_sessions** 的内存是在启动时分配的，所以，如果将 **min_sessions** 的数量设置得非常高（以便允许大量用户连接），则大部分内存将被增强型全文本搜索引擎的用户连接独占。

通过将 **min_sessions** 设置为将要使用的用户会话的平均数，可以提高增强型全文本搜索引擎的性能。这样可避免增强型全文本搜索引擎在用户会话开始时就必须分配内存。

使用 `sp_text_notify`

在决定以多大的频率发出 `sp_text_notify` 之前，应了解站点的需求。

使用 `sp_text_notify` 会在增强型全文本搜索引擎上产生一个负载，原因是该系统过程会读取数据并更新文本集合。发出 `sp_text_notify` 可能会对性能造成很大的影响，具体取决于此负载的大小。由于牵涉到性能，因此您必须决定索引更新的频率。如果索引必须尽可能保持最新，则需要频繁发出 `sp_text_notify`（每 5 秒就需要发出一次）。但是，如果索引并不是迫切需要更新，则最好等到系统活动不那么频繁时再发出 `sp_text_notify`。

注释 不能从事务内部发出 `sp_text_notify`。

配置多个增强型全文本搜索引擎

对于使用频繁的表，可将这些表的文本索引放到不同的增强型全文本搜索引擎上以提高性能。因为用户可以将查询分布到多个增强型全文本搜索引擎上，而不是将所有的查询发送到一个引擎上，所以可以提高性能。每个 Adaptive Server 都可以连接到多个增强型全文本搜索引擎，但每个增强型全文本搜索引擎只能连接到一个 Adaptive Server。

启动时创建多个增强型全文本搜索引擎

如果开始时创建了多个增强型全文本搜索引擎，则可以编辑 `installtextserver` 脚本，以便包含所有这些增强型全文本搜索引擎。请参见第 13 页的“[编辑 installtextserver 脚本](#)”。

添加增强型全文本搜索引擎

通过从 isql 发出 `sp_addserver`，可在完成初始启动后添加增强型全文本搜索引擎：

```
sp_addserver server_name [, server_class [,  
physical_name]]
```

其中：

- *server_name* 是用来标识系统上的服务器（本例中为增强型全文本搜索引擎）的名称。
- *server_class* 标识要添加的服务器的类别。对于增强型全文本搜索引擎，该值为“sds”。
- *physical_name* 是 *server_name* 服务器在 `interfaces` 文件中所使用的名称。

有关详细信息，请参见《参考手册》中的 `sp_addserver`。

例如，若要添加名为 BLUE 的增强型全文本搜索引擎，请输入：

```
sp_addserver BLUE, sds, BLUE
```

在配置并启动增强型全文本搜索引擎之后，可以使用以下语法查看是否可从 Adaptive Server 连接到该增强型全文本搜索引擎：

```
server_name...sp_show_text_online
```

例如，若要连接到名为 BLUE 的服务器，请输入：

```
BLUE...sp_show_text_online
```

配置其它增强型全文本搜索引擎

每个增强型全文本搜索引擎都需要有自己的以下各项：

- `interfaces` 文件条目
- 配置文件

所有增强型全文本搜索引擎都使用相同的数据库（缺省情况下名为 `text_db`）来存储文本索引元数据，并使用相同的 `vesaux` 和 `vesauxcol` 表。

多用户

以下提示可帮助避免因多用户引起的死锁：

- 1 确保 Adaptive Server 使用与增强型全文本搜索相同的连接数。缺省值为 100。

```
sp_configure "user connections", 100
```

- 2 确保 `vesaux`, `vesauxcol` 和 `text_events` 表（在 `model` 数据库或每个新数据库中）使用行级锁定。

对于现有表：`alter table table_name lock datarows`

对于新表：`create table ... lock datarows`

- 3 对于大批命令，可尝试将其分为较小的事务。
- 4 如果仍然出现死锁，可增加 Adaptive Server 可用的锁数，并调整行锁升级设置。请参见《系统管理指南》。

文件描述符和增强型全文本搜索

在执行搜索时，增强型全文本搜索会大量使用文件描述符。对于并发搜索和大文本索引，这可能会导致连接时收到以下错误消息：

```
ERRORMSG, Error (): Available files (-1) less than min 5
```

此消息表示增强型全文本搜索进程已经用完了对该进程限定的文件描述符。如果出现此错误消息，可增大增强型全文本搜索进程的文件描述符的限制数量。

`max_session_fd` 可限制增强型全文本搜索会话可以分配的文件描述符的数量。在无法增大文件描述符的限制数量时，可以使用该参数。

`max_session_fd` 的缺省值为 0，这表示每个会话只受增强型全文本搜索进程文件描述符的限制数量的限制。`max_session_fd` 的最小设置为 5。

如果出现了上面的错误消息，但无法增大增强型全文本搜索的文件描述符的限制数量，那么，最好是按以下公式来测算该参数应设置的值：
(最大文件描述符数 - 20) / (最大并发连接数)。

例如，如果文件描述符的限制数量设为 1024，与增强型全文本搜索的最大并发连接数是 50，则计算方法如下：

$$((1024) - 20)/50 \approx 20$$

增强型全文本搜索会话可用的文件描述符越多越好。

因为每个增强型全文本搜索会话使用的文件描述符的最大数量受到限制，所以性能可能会有所下降。Sybase 建议应谨慎使用 `max_session_fd`。必须重新启动增强型全文本搜索进程，才能使 `max_session_fd` 更改生效。

Verity 主题

本章是经过 Verity 的许可而翻印的。目的是使增强型全文本搜索用户能够对 Verity 主题有所了解。

主题	页码
主题的定义	81
使用主题大纲文件	82
准备主题	83
主题知识库	83
主题的结构	85
最大主题数	88
Verity 查询语言	88
主题大纲示例	93
运算符参考	94
修饰符参考	103
权值和文档重要性	104
主题得分和文档重要性	109
设计主题	111
准备主题设计	112
主题设计策略	114
设计初始主题	115

主题的定义

主题是与某个概念或科目范围相关的一组信息。主题提供了一种概述知识的便利手段，并可作为共享资源供最终用户使用。通过在 Verity 应用程序中添加主题，用户可以更轻松地为主题所代表的事项上进行搜索。

可以将主题组合起来构成一个知识库来代表一个知识分类，用户在执行搜索时可在知识分类中查找。知识库为用户提供了无须使用复杂的语法编写繁琐的查询即可查找所需信息的能力。

主题组织

主题按照类似于大纲的格式将相关搜索条件组织成组。运算符和修饰符起着粘合剂的作用，用以将相关的搜索条件组连结在一起。在创建主题时，既可以将其作为独立的单元，也可以将其作为与层次结构中的其它主题存在一定关系的单元。

权值分配

在一个主题结构中，可以为一些搜索条件组提供比其它搜索条件组更高的权值。为搜索条件分配权值会影响搜索中选取的文档的重要性；一个文档离结果列表顶部越近，则说明该文档对此搜索条件越重要或相关程度更高。搜索条件权值是一个介于 0.01 和 1.00 之间的数字。所选文档在结果列表中的位置，有助于大致确定该文档相对于搜索条件的相关程度。

使用主题大纲文件

可以通过创建主题大纲文件来编写主题。

主题大纲文件是采用结构化格式的 ASCII 文本文件，其中包含主题定义。主题大纲文件可能具有如下形式：

```
$Control:1
art <Accrue>
*performing-arts <Accrue>
**0.80 "ballet"
**0.50 "drama"
**0.50 'dance'
**0.80 "opera"
**0.80 "symphony"
**0.90 "chamber music"
**"Isaac Stern"
*film <Accrue>
**directors <Filter>
/definition="title CONTAINS Truffaut"
*visual-arts <Accrue>
literature <Accrue>
philosophy <Accrue>
language <Accrue>
history <Accrue>
$$
```

可以使用任何文本编辑器来创建主题大纲文件。

准备主题

可供用户使用的主题必须存在于一个使用 `mktopics` 实用程序生成的主题组中。用 `mktopics` 生成的 Verity 主题组可供任何 Verity 应用程序使用。一个主题组最多可支持 20,000 个主题定义。一个主题组所允许的确切主题数取决于用于定义主题的 Verity 查询语言。

设置过程

准备主题以供用户使用的过程分为三个步骤，概括如下：

- 1 使用主题大纲文件创建主题定义。
- 2 生成主题组。可以使用 `mktopics` 实用程序创建主题组。`mktopics` 实用程序创建主题组，还可在特定集合上编制主题索引。
- 3 将主题组导入到增强型全文本搜索引擎。

主题知识库

本部分讨论知识库的主要功能，以及用于为知识库定义主题的组织格式。包括主题知识库的以下方面：

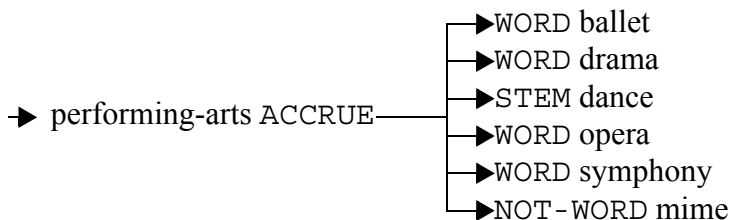
- 将主题组合到知识库中
- 主题的结构
- 主题与子主题之间的关系
- 主题类型
- 主题的命名

将主题组合到知识库中

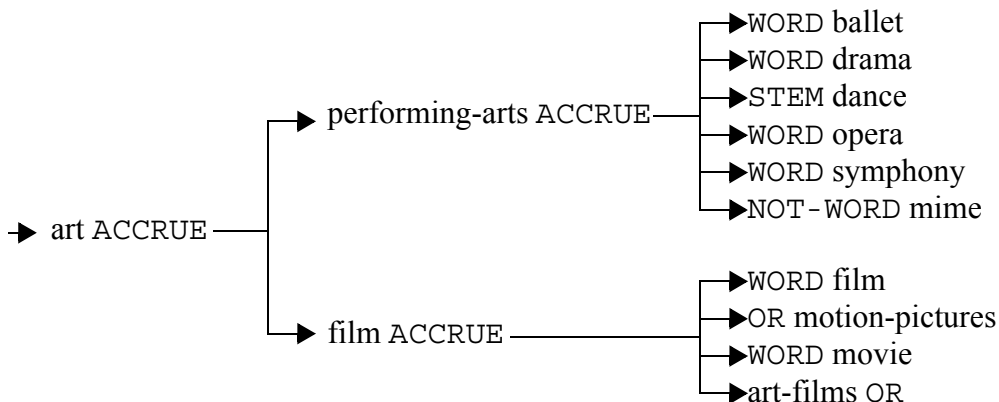
主题是与某个概念或科目范围相关的一组信息。知识库是一组称为主题的概念。将主题组合到知识库中能使用户方便地查找已保存为主题的概念。

主题的科目范围一般由主题的名称来标识。在下面示例中，主题的对象是 *performing-arts*。该主题由两个结构元素组成，即它的名称 *performing-arts* 和它的实证主题：*ballet*、*musical*、*dance*、*opera*、*symphony* 和 *drama*。

运算符和修饰符起着粘合剂的作用，它们可以将相关的实证主题连结在一起。运算符表示要对实证主题应用的逻辑。此逻辑定义要查找的文档种类的限定条件。修饰符进一步对实证主题应用逻辑。例如，修饰符可以指定不在结果列表中包括包含某个实证主题的文档。



在添加了主题之后，主题的结构将变得更加复杂。在下面的示例中，已将主题 *film* 添加到结构中，以构成目前的顶层主题 *art*。在此结构中，*performing-arts* 和 *film* 是主题 *art* 的子主题。



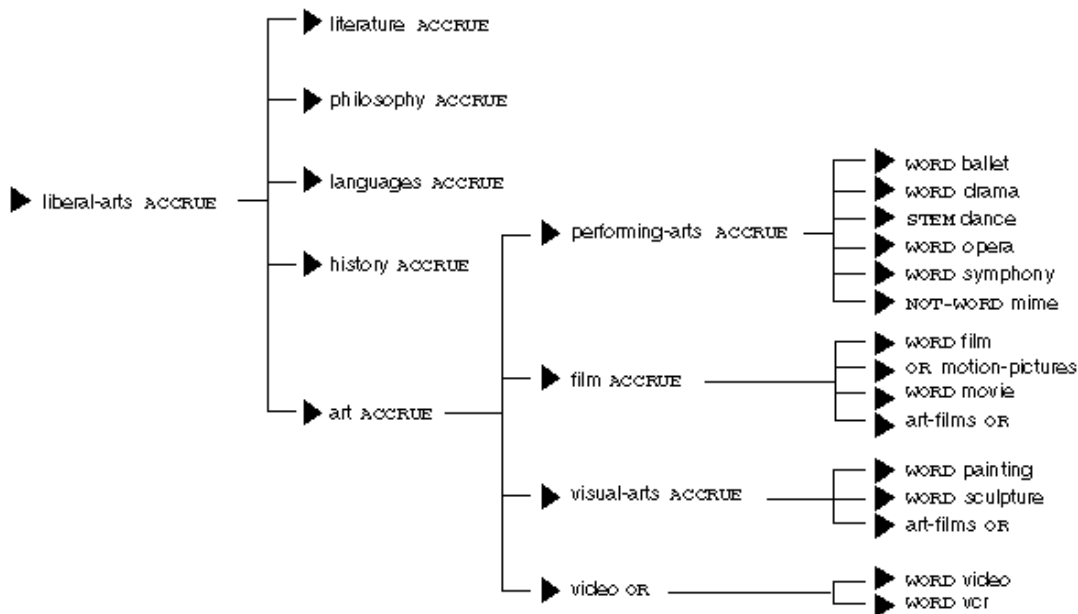
复杂主题由顶层主题、子主题和实证主题组成。这几个元素可确定一个主题的相关科目范围。知识库通常由几个顶层主题组成。子主题和实证主题可由多个顶层主题使用。

主题的结构

在处理搜索时，主题的结构会影响解释主题的方式。为了将主题设计为准确地表达某个概念，需要使用下述组件来定义主题结构。

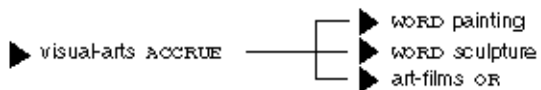
顶层主题

顶层主题是在主题结构中定义的最高层主题。顶层主题代表希望 Verity 搜索代理查找的科目范围。在下面的示例中，可以考虑将 literature、philosophy、languages、history 和 art 作为顶层子主题，它们构成了顶层主题 liberal-arts。



子主题

子主题构成了介于顶层主题与实证主题之间的层次。子主题的名称应表明由其子主题或实证主题共同描述的科目范围。例如，子主题 `visual-arts` 包括几个相关单词或实证主题，如下所示：

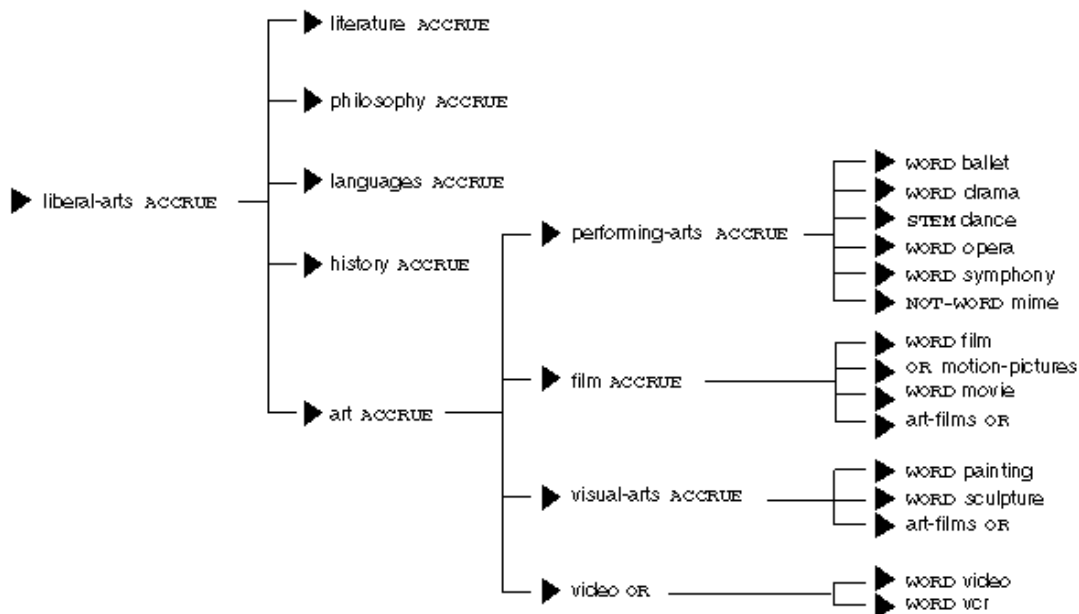


实证主题

实证主题是主题结构中的最下层单元。实证主题是由字母数字字符组成的字符串。一个实证主题最多可包含 128 个字母数字字符。

主题和子主题的关系

每个主题与其关联子主题构成了一个层次式父子关系。在下面的示例中，子主题 `performing-arts`、`film`、`visual-arts` 和 `video` 是 `art` 主题的子项。`art` 主题本身是 `liberal-arts` 主题的一个子项。反之，`liberal-arts` 主题也可以是该结构中邻近的更高层父主题的子项。



当使用一个主题执行搜索时，由该主题定义的科目范围包括其子主题、子主题的子主题，等等，一直到结构中的实证主题。不属于所用主题的直接子项的主题将不会包括在搜索中。

例如在上面的示例中，使用 `film` 主题的搜索将使 Verity 搜索引擎查找包含有关 `film`、`motion pictures`、`movies` 和 `art films` 的信息的文档。在本示例中，搜索引擎将不查找与 `performing-arts`、`visual-arts` 或 `video` 主题相关的文档，因为这些主题不是 `film` 主题的子项。但是，如果使用 `art` 主题，搜索时将查找与 `art` 主题的所有子项（包括 `performing-arts`、`film`、`visual-arts` 和 `video`）相关的文档。

最大主题数

代表一个知识库的单个主题组可由多达 20,000 个的主题组成，其中包括顶层主题、子主题和实证主题。在搜索中使用包含多达 1,000 个子主题的主题时，可能会超出内存限制。

主题命名问题

- 主题名称最多可包含 128 个字母数字字符，包括连字符和下划线。
- 主题名称和实证主题通常区分大小写。命名实证主题时，可以全部用大写字母，如 APPLE，可采用首字母大写，如 Apple，或者全部用小写，如 apple。执行搜索时将不会考虑大小写。因此，如果输入 APPLE 作为实证主题，Verity 搜索引擎将选择包含 “APPLE”、“Apple” 或 “apple” 的文档。

但可以使用 CASE 修饰符来指定与实证主题条目匹配时要区分大小写。

Verity 查询语言

本部分介绍 Verity 查询语言。Verity 查询语言由可以用来创建主题的运算符和修饰符组成。运算符代表要对搜索元素应用的逻辑，可以对搜索元素进行组合以创建主题。此逻辑定义要查找的文档种类的限定条件。修饰符进一步对搜索元素应用逻辑。例如，修饰符可以指定搜索元素是区分大小写的。

本部分包括下列内容：

- 查询语言摘要
- 运算符优先规则
- 主题大纲示例
- 运算符参考
- 修饰符参考

查询语言摘要

Verity 查询语言由运算符和修饰符组成。运算符和修饰符都代表要对搜索元素应用的逻辑。此逻辑定义了限定条件，文档必须满足这些条件才能被检索出来。运算符按其类型进行分类，如下所述：

- 实证运算符
- 邻近运算符
- 关系运算符
- 概念运算符
- 布尔运算符

修饰符扩展运算符所应用的逻辑，并与运算符进行组合使用。

实证运算符

实证运算符将搜索单词扩展成一个相关单词列表，然后还将对该列表进行搜索。当使用实证运算符执行搜索时，包含扩展单词列表中单词的一个或多个实例的文档也就是包含所指定的单词及其同义词的文档。使用实证运算符检索出的文档不按相关性进行排列，除非您使用了 MANY 修饰符。请参见第 103 页的“MANY 修饰符”。表 8-1 描述了每个实证运算符。

表 8-1: 实证运算符

运算符名称	说明
WORD	选择包含指定的单词的一个或多个实例的文档。
STEM	选择包含所指定的搜索单词的一种或多种变化形式的文档。
THESAURUS	选择包含指定的单词的一个或多个同义词的文档。
WILDCARD	选择包含某个含有变量的字符串的匹配项的文档。
SOUNDEX	选择包含一个或多个与指定字“发音相像”或拼写相似的字的文档。
NEAR/N	扩展搜索以包括所输入的单词以及类似于查询词条的单词。此运算符执行“近似模式匹配”以标识类似的字。

邻近运算符

邻近运算符指定特定的单词在文档中的相对位置，也就是说，指定的单词必须在要检索的文档的同一个短语、段落或句子中。在使用 NEAR 和 NEAR/N 运算符的情况下，将按照指定的单词的邻近程度对检索到的文档进行相关性排列。当嵌套使用邻近运算符时，应首先使用范围最宽的运算符，即短语或单个单词可以出现在 SENTENCE 或 PARAGRAPH 运算符内，而 SENTENCE 运算符可以出现在 PARAGRAPH 运算符内。

表 8-2 描述了每个邻近运算符。

表 8-2: 邻近运算符

运算符名称	说明
IN	选择在一个或多个文档区域中包含指定值的文档。文档区域代表文档的某一个区域，例如文档的摘要、日期或正文文字。
PHRASE	选择包含所指定的短语的文档。短语是以特定顺序出现的两个或更多个单词的组合。
SENTENCE	选择在一个句子内包含所指定的全部单词的文档。
PARAGRAPH	选择在一个段落内包含所指定的全部搜索元素的文档。
NEAR	选择包含指定的搜索词条且这些词条彼此邻近的文档。
NEAR/N	选择包含两个或更多个彼此相隔距离在 N 个单词内的搜索单词的文档，其中 N 为整数。

关系运算符

关系运算符可搜索已在集合中定义的文档字段（如 AUTHOR）。这些运算符通过选择包含指定字段值的文档来执行过滤功能。关系运算符使用的字段可以包含字母数字字符。使用关系运算符检索的文档不按相关性排列，并且不能对关系运算符使用 MANY 修饰符。

在创建主题时，关系运算符总是与特殊的 FILTER 运算符结合使用。有关正确语法，请参见本节后面的“主题大纲示例”中“visual-arts”主题下的示例。

有多种关系运算符可用于进行数字和日期比较，其中包括：=（等于），>（大于）、>=（大于或等于）、<（小于）以及 <=（小于或等于）。

有多种关系运算符可用于进行文本比较，其中包括：

表 8-3：关系运算符

运算符名称	说明
CONTAINS	通过将指定的单词或短语与特定文档字段中存储的值进行匹配来选取文档。
MATCHES	通过将指定的字符串与特定文档字段中存储的值进行匹配来选取文档。
STARTS	通过将指定的字符串与特定文档字段中存储的值的开头若干个字符进行匹配来选取文档。
ENDS	通过将指定的字符串与特定文档字段中存储的值的结尾若干个字符进行匹配来选取文档。
SUBSTRING	通过将指定的字符串与特定文档字段中存储的值的部分字符串进行匹配来选取文档。

概念运算符

概念运算符组合搜索元素的含义，以便在文档中识别概念。使用概念运算符检索出的文档按相关性排列。[表 8-4](#) 描述了每个概念运算符。

表 8-4：概念运算符

运算符名称	说明
AND	选取包含所指定的全部搜索元素的文档。
OR	选取至少显示一个搜索元素的实证的文档。
ACCRUE	选取至少包含所指定的一个搜索元素的文档。

布尔运算符

可以为主题分配布尔运算符，以检索包含该主题的任意一个或全部子项的文档。与使用概念运算符创建的主题不同，布尔运算符不接受权值。[表 8-5](#) 描述了每个概念运算符。

表 8-5：布尔运算符

运算符名称	说明
ALL	选取包含一个主题的全部子项的文档。
ANY	选取至少包含某个主题的一个子项的文档。

修饰符

修饰符影响运算符的行为。表 8-6 描述了每个修饰符。

表 8-6: 修饰符

运算符名称	说明
CASE	执行区分大小写的搜索。
MANY	计算文档中的单词或短语的密度，并为所检索的文档生成一个相关性排列得分。
NOT	排除显示指定单词或短语的实证的文档。
ORDER	指定搜索元素必须遵从的出现顺序。

运算符优先规则

Verity 搜索引擎使用优先规则确定如何分配运算符。这些规则指出，在分配给主题时，某些运算符的等级高于其它运算符，并且会影响执行文档选取的方式。

表 8-7 描述了如何对运算符应用优先规则。

表 8-7: 优先规则

运算符	优先级	如何确定优先级
AND OR ACCRUE	最高优先级	概念运算符的优先级高于其它运算符。因此，对于使用这些运算符的主题的子主题，可以赋予在“增量优先级”或“最低优先级”下列出的任何运算符。
ALL PARAGRAPH SENTENCE NEAR NEAR/N PHRASE ANY	递增优先级 (按递减顺序)	邻近运算符引用文档内存在的递增范围。可以为使用这些运算符主题的子主题分配优先级次序中次最低级运算符。因此，短语优先于单词，句子优先于短语和单词，段落优先于句子、短语和单词。
WORD STEM SOUNDEX WILDCARD THESAURUS	最低优先级	实证运算符处于主题结构中的最低层。由于实证运算符用于文档中包含的单词，因此这些运算符都具有相同的优先级。

为了避免优先级冲突，不要在子主题包含概念运算符（AND、OR、ACCURUE）的父主题中使用 ANY 或 ALL。不能为使用 ANY 或 ALL 的主题分配可变权值，因此不能在子主题允许可变权值（AND、OR、ACCURUE）的父主题中使用这些运算符。使用 ANY 和 ALL 限制了只能对是否存在进行评估（得分为 0.00 或 1.00）。如果满足条件，这些主题的子项将自动获得 1.00 的得分；如果不满足条件，这些主题的子项将自动获得 0.00 的得分；因此为这些子项分配可变权值（如 0.80）没有意义。

主题大纲示例

下列主题与用主题大纲文件创建的主题相同：

```
$Control:1
art <Accrue>
*performing-arts <Or>
**0.80 "drama"
**0.50 "theater"
**0.80 'dance'
*film <And>
**0.90 "cinema"
**0.90 "documentary"
**newsreel <Filter>
/definition="DATE >= 05/01/96"
*film-makers <Accrue>
**"Woody Allen"
*film-making <Paragraph>
**"direct"
**"produce"
*visual-arts <Accrue>
**sculpture <In>
/zonespec="title"
**painters <Filter>
/definition="Title MATCHES Famous Painters"
**<Thesaurus>
/wordtext="paint"
literature <Accrue>
*novels <Near>
**0.80 "Proust"
**0.80 "Remembrance" <Case>
*american-novel <Sentence>
**"American"
**"novel"
```

```
history <Accrue>
*<Wildcard>
/wordtext="histor*"
music <Accrue>
*jazz
**"bebop"
**<Not> "fusion"
*classical
**"Italian opera"
$$
```

运算符参考

下面按字母顺序列出了每个运算符。在上一部分的主题大纲中，可以找到其中许多运算符的示例。

ACCRUE 运算符

选取至少包含所指定的一个搜索元素的文档。有效搜索元素为两个或更多个单词或短语。选取的文档按相关性进行排列。

ACCRUE 运算符按照文档中出现的每个搜索元素对选取的文档评分，评分方法依据“越多越好”原则：文档中发现的搜索元素越多，文档的得分就越高。[第 93 页的“主题大纲示例”](#)的大纲文件示例中有 ACCRUE 运算符的多个示例。

ALL 运算符

选取包含所指定的全部搜索元素的文档。与 ACCRUE 运算符不同，使用 ALL 运算符时不能分配权值。

AND 运算符

选取包含所指定的全部搜索元素的文档。使用 AND 运算符选取的文档按相关性进行排列。“主题大纲示例”中的示例说明了如何将 AND 运算符与主题“film”一起使用。在该示例中，只有同时包含搜索单词和大于或等于 05/01/96 的日期的那些文档才能被选中并按得分进行排列。

ANY 运算符

选取至少包含所指定的一个搜索元素的文档。与 ACCRUE 运算符不同，使用 ANY 运算符时不能分配权值。

CONTAINS 运算符

通过将指定的单词或短语与特定文档字段中存储的值进行匹配来选取文档。使用 CONTAINS 运算符时，需要指定要搜索的字段名和要搜索的单词或短语。

使用 CONTAINS 运算符时，存储在文档字段中的单词被解释为单个的顺序单元。可以指定一个或多个这些单元作为搜索条件。如果要指定多个单词，则每个单词必须顺序相连，并且必须用空格分隔。将 CONTAINS 与 FILTER 运算符配合使用。

CONTAINS 语法与 MATCHES 语法相同。请参见第 93 页的“主题大纲示例”中“visual arts”主题下的 MATCHES 示例。此示例假定已为集合创建了 TITLE 字段。

CONTAINS 运算符不能识别非字母数字字符。CONTAINS 运算符将非字母数字字符解释为空格，并将所分隔的值当作独立单元进行处理。例如，如果您已定义 (/) 为有效字符，并输入包含该字符的搜索条件，如 OS/2，那么“OS”和“2”将被当作独立单元来处理。

CONTAINS 运算符不引用 *style.lex* 文件来定义一个单词中包括哪些字符。

ENDS 运算符

通过匹配所指定的字符串来选取文档。将 ENDS 与 FILTER 运算符配合使用。ENDS 的语法与 MATCHES 的语法相同。请参见第 93 页的“主题大纲示例”中“visual arts”主题下的 MATCHES 示例。此示例假定已为集合创建了 TITLE 字段。

= (EQUALS) 运算符

选取文档字段值与所指定的搜索字符串完全相同的文档。将 EQUALS 与 FILTER 运算符配合使用。EQUALS 的语法与 GREATER THAN OR EQUAL TO 的语法相同。请参见第 93 页的“主题大纲示例”中“film”主题下的 GREATER THAN OR EQUAL TO 示例。此示例假定已为集合创建了 DATE 字段。

FILTER 运算符

此特殊 FILTER 运算符与关系运算符结合使用来进行字段搜索。有关正确语法，请参见第 93 页的“主题大纲示例”中“visual-arts”主题下的示例。

> (GREATER THAN) 运算符

选取文档字段值大于所指定的搜索字符串的文档。将 GREATER THAN 与 FILTER 运算符配合使用。GREATER THAN 的语法与 GREATER THAN OR EQUAL TO 的语法相同。请参见第 93 页的“主题大纲示例”中“film”主题下的 GREATER THAN OR EQUAL TO 示例。此示例假定已为集合创建了 DATE 字段。

>= (GREATER THAN OR EQUAL TO) 运算符

选取文档字段值大于或等于所指定的搜索字符串的文档。将 GREATER THAN OR EQUAL TO 与 FILTER 运算符配合使用。请参见第 93 页的“主题大纲示例”中“film”主题下的示例。此示例假定已为集合创建了 DATE 字段。

< (LESS THAN) 运算符

选取文档字段值小于所指定的搜索字符串的文档。将 LESS THAN 与 FILTER 运算符配合使用。LESS THAN 的语法与 GREATER THAN OR EQUAL TO 的语法相同。请参见第 93 页的“主题大纲示例”中“film”主题下的 GREATER THAN OR EQUAL TO 示例。此示例假定已为集合创建了 DATE 字段。

<= (LESS THAN OR EQUAL TO) 运算符

选取文档字段值小于或等于所指定的搜索字符串的文档。将 LESS THAN OR EQUAL TO 与 FILTER 运算符配合使用。LESS THAN OR EQUAL TO 的语法与 GREATER THAN OR EQUAL TO 的语法相同。请参见第 93 页的“主题大纲示例”中“film”主题下的 GREATER THAN OR EQUAL TO 示例。此示例假定已为集合创建了 DATE 字段。

IN 运算符

选择在一个或多个文档区域中包含指定值的文档。文档区域代表文档的某一个区域，例如文档的摘要、日期或正文文字。只有在集合中定义了文档区域后，IN 运算符才起作用。如果使用 IN 运算符来搜索未定义区域的集合，将不会选取任何文档。此外，所指定的区域名必须与集合中定义的区域名相匹配。请向集合管理员咨询，确定已为特定集合定义了哪些区域。第 93 页的“主题大纲示例”中的示例说明了如何将 IN 运算符与单词 “sculpture” 和 TITLE 区域一起使用。

MATCHES 运算符

通过将指定的字符串与特定文档字段中存储的值进行匹配来选取文档。当使用 MATCHES 运算符时，应指定要搜索的字段名称和要搜索的单词、短语或数字。

与 CONTAINS 运算符不同的是，对 MATCHES 运算符指定的搜索条件必须与要选取的文档的字段值完全匹配。对于 MATCHES 运算符，将不选取作为值的一部分出现的任意搜索字符串匹配项，而只选取与整个搜索字符串匹配的值。

可以使用问号 (?) 来代表字符串内的单个可变字符，使用星号 (*) 来匹配字符串内的多个可变字符。

将 MATCHES 与 FILTER 运算符配合使用。第 93 页的“主题大纲示例”中的示例说明了如何将 MATCHES 与短语 “famous painters” 和字段 TITLE 一起使用。此示例假定已为集合创建了 TITLE 字段。

NEAR 运算符

选择包含指定的搜索词条且这些词条彼此邻近的文档。文档得分将根据搜索词条之间的相对字数来计算。例如，如果搜索表达式包含两个字，且这些字在文档中彼此相邻出现（故区域大小是两个字长），那么为该文档赋予的得分是 1.00。因此，在其中最小的区域内包含所有搜索术语的文档总是获得最高分。搜索词条彼此相隔不在 1000 个单词内的文档将不被选取，因为搜索词条可能离得太远，在文档的上下文内已没有什么意义。

使用 NEAR 运算符只能查找彼此相隔距离很近的搜索单词，从这个意义上来说，它与其它邻近运算符类似。但是，与其它邻近运算符不同的是，NEAR 运算符计算相对邻近度，并根据计算来分配得分。

第 93 页的“主题大纲示例”中的示例说明了如何将 NEAR 与主题“novels”一起使用。

NEAR/N 运算符

选择包含两个或更多个彼此相隔距离在 N 个单词内的搜索单词的文档，其中 N 为整数。当指定的单词由 N 个或少于 N 个的单词分隔时，将基于指定单词的相对距离来计算文档得分。将不选取包含指定单词、但这些单词由 N 个以上的单词分隔的文档。例如，如果搜索表达式 NEAR/5 用于查找两个彼此相隔距离在五个单词内的单词，那么如果有两个文档均满足此条件，但一个文档包含的指定单词的相隔距离在三个单词内，另一个文档包含的指定单词的相隔距离在五个单词内，那么前者的得分将高于后者。

变量 N 可以是介于 1 和 1,024 之间的整数，其中 NEAR/1 将搜索彼此相邻的两个单词。如果 N 是 1,000 或以上，则在指定该值时一定要使用逗号，例如 NEAR/1000。

使用 NEAR/N 运算符只能查找彼此相隔距离很近的搜索单词，从这个意义上来说，它与其它邻近运算符类似。但是，与其它邻近运算符不同的是，NEAR/N 运算符基于相对邻近度来分配得分。

OR 运算符

选取至少显示一个搜索元素的实证的文档。使用 OR 运算符选取的文档按相关性进行排列。第 93 页的“主题大纲示例”中的示例说明了如何将 OR 与主题“performing-arts”一起使用。

PARAGRAPH 运算符

选择在一个段落内包含所指定的全部搜索元素的文档。有效搜索元素为两个或更多个单词或短语。可以按连续顺序或随机顺序指定搜索元素。只要文档包含出现在同一段落中的搜索元素，就会被检索到。第 93 页的“主题大纲示例”中的示例说明了如何将 PARAGRAPH 与“film-making”主题一起使用。

PHRASE 运算符

选择包含所指定的短语的文档。短语是以特定顺序出现的两个或更多个单词的组合。在实证字段中输入多个单词时，必须使用 PHRASE 运算符。使用 PHRASE 运算符的单词显示在双引号内。第 93 页的“主题大纲示例”中所示的“Woody Allen”和“Italian opera”使用了 PHRASE 运算符。

SENTENCE 运算符

选择在一个句子内包含所指定的全部单词的文档。可以按连续顺序或随机顺序指定搜索元素。只要文档包含出现在同一句中的搜索元素，就会被检索到。第 93 页的“主题大纲示例”中的示例说明了如何将 SENTENCE 与“american-novel”主题一起使用。

SOUNDEX 运算符

选择包含一个或多个与指定字“发音相像”或拼写相似的字的文档。只有首字母与指定单词的首字母相同的单词才能被选取。例如，当与“sale”一起使用 SOUNDEX 时，所选取的文档将包含诸如“sale”、“sell”、“seal”、“shell”、“soul”和“scale”这样的单词。只有当使用了 MANY 修饰符时，才会对文档进行相关性排列。

STARTS 运算符

通过将指定的字符串与特定文档字段中存储的值的开头若干个字符进行匹配来选取文档。将 STARTS 与 FILTER 运算符配合使用。STARTS 的语法与 MATCHES 的语法相同。请参见第 93 页的“主题大纲示例”中“visual arts”主题下的 MATCHES 示例。此示例假定已为集合创建了 TITLE 字段。

STEM 运算符

选择包含所指定的搜索单词的一种或多种变化形式的文档。与 STEM 运算符一起使用的单词显示在单引号内。在“主题大纲示例”的示例中，对单词“dance”使用了 STEM 运算符。因此所选取的文档将包含如“dances”、“danced”、“dancing”以及“dance”等单词。

SUBSTRING 运算符

通过将指定的字符串与特定文档字段中存储的值的部分字符串进行匹配来选取文档。组成字符串的字符可出现在字段值的开头、中间或末尾。SUBSTRING 的语法与 MATCHES 的语法相同。请参见“主题大纲示例”中“visual arts”主题下的 MATCHES 示例。此示例假定已为集合创建了 TITLE 字段。

THESAURUS 运算符

选择包含指定的单词的一个或多个同义词的文档。例如，当与单词“altitude”一起使用 THESAURUS 运算符时，所选取的文档将包含诸如“height”和“elevation”这样的单词。只有当使用了 MANY 修饰符时，才会对文档进行相关性排列。

TYPO/N 运算符

TYPO/N 运算符将搜索扩展为包括指定单词及其类似单词。*N* 变量可选，它指定查询词条与匹配词条之间的最大错误数。例如，TYPO mouse 返回包含如“house”、“louse”和“moose”等单词的文档。

WILDCARD 运算符

选择包含某个含有变量的字符串的匹配项的文档。WILDCARD 运算符允许定义含有变量的搜索字符串，可用于在文档中查找相关单词的匹配。第 93 页的“主题大纲示例”中的示例说明了如何使用字符串“histor*”搜索“history”、“historical”和“historian”等单词。只有当使用了 MANY 修饰符时，才会对文档进行相关性排列。

使用特殊通配符

可以对 WILDCARD 运算符使用下列通配符来表示搜索字符串中的可变部分。

表 8-8: 特殊通配符

字符	作用
?	指定任意一个字母数字字符，例如 ?an，可查找 “ran”、“pan”、“can” 和 “ban”。使用问号时，不必指定 WILDCARD 运算符。在集合 ([]) 或可选模式 ({ }) 中，问号将被忽略。
*	指定零个或多个字母数字字符，例如在 corp* 中，将查找 “corporate”、“corporation”、“corporal” 和 “corpulent” 等。使用星号时，不必指定 WILDCARD 运算符，并且不能使用星号指定通配符字符串的第一个字符。在集合 ([]) 或可选模式 ({ }) 中，星号将被忽略。
[]	指定一个集合中的任意一个字符，如在 <WILDCARD> ‘c[auo]t’ 中，将查找 “cat”、“cut” 和 “cot”。注意，必须将包含集合的单词用反引号 (`) 括起来，且集合中不能有空格。
{ }	指定由逗号分隔的多个模式中的一个，如在 <WILDCARD> ‘bank{s,er,ing}’ 中，将查找 “banks”、“banker” 和 “banking”。注意，必须将包含模式的单词用反引号 (`) 括起来，且集合中不能有空格。
^	指定不在集合中的任一个字符，如在 <WILDCARD> ‘st[^oa]ck’ 中，将排除 “stock” 和 “stack”，但会查找 “stick” 和 “stuck”。注意，尖号 (^) 必须是引入集合的左括号 ([) 后的第一个字符。
-	指定集合中的一个字符范围，如在 <WILDCARD> ‘c[a-r]t’ 中，将查找从 “cat” 到 “crt” 的每个三个字母单词。

查找非字母数字字符

请记住，只有已建立了用于创建搜索集合的 *style.lex* 文件以识别要搜索的字符，才能搜索非字母数字字符。有关信息请咨询集合管理员。

将通配符作为文字进行搜索

以上列出的通配符都解释为通配符而非文字字符，除非采用反斜杠 (\) 将它们分隔开来。如果要将通配字符串中的通配符解释为文字，则必须在该字符之前加上反斜杠。例如，要匹配通配字符串中的文字星号 (*)，可用下列形式分隔该字符：

```
<WILDCARD> a\*
```

将特殊字符作为文字进行搜索

下列非字母数字字符执行特殊的内部功能，且在缺省情况下，在通配字符串中不将其当作文字。

- 逗号,
- 左、右圆括号 ()
- 双引号 “
- 反斜杠 \
- at 号 @
- 左大括号 {
- 左方括号 [
- 小于号 <
- 反引号 `

如果要将特殊字符解释为文字，必须用反引号 (`) 括住整个通配字符串。例如，若要搜索通配字符串 “a{b”，请用反引号按如下方式括住该字符串：

```
<WILDCARD> `a{b`
```

要搜索包含文字反引号字符 (`) 的通配字符串，必须连续使用两个反引号 (``)，并将整个通配字符串括在反引号 (`) 内，如下所示：

```
<WILDCARD> ``n``t`
```

如果已建立了用于创建搜索集合的 *style.lex* 文件来识别反引号字符，则只能搜索反引号字符。有关信息请咨询集合管理员。

WORD 运算符

选择包含指定的单词的一个或多个实例的文档。含有 WORD 运算符的单词显示在双引号内。第 93 页的“主题大纲示例”中的示例显示了 WORD 运算符的许多实例。

修饰符参考

修饰符进一步指定运算符的行为。例如，可以对运算符使用 CASE 修饰符，以指定将所输入的搜索单词的大小写也视为一个搜索元素。修饰符包括 CASE、MANY、NOT 和 ORDER，其描述如下。

CASE 修饰符

对 WORD 或 WILDCARD 运算符使用 CASE 修饰符，可根据所指定的单词或短语的大小写执行区分大小写的搜索。

缺省情况下，将检索出所有包含搜索单词或短语的文档，而不考虑其大小写。要使用 CASE 修饰符，只需按照希望它在检索文档中出现的形式输入搜索单词或短语即可 – 可以全部用大写字母、混用大小写字母，也可以全部用小写字母。第 93 页的“主题大纲示例”中的示例说明了如何对 CASE 修饰符使用单词“Remembrance”，以引用 Proust 的小说名 *Remembrance of Things Past* 中的第一个字。

MANY 修饰符

计算文档中的字、派生变形或短语的密度，并为检索到的文档生成一个相关性排列得分。字、派生变形或短语的出现次数相对于文档文本量的比例越高，检索时该文档的得分就越高。由于 MANY 修饰符考虑相对于文档文本的密度，因此某个单词的出现次数较多的较长文档的得分可能低于该单词的出现次数较少的较短文档。

MANY 修饰符可以与下列运算符一起使用：WORD、WILDCARD、STEM、SOUNDEX、PHRASE、SENTENCE、PARAGRAPH 和 THESAURUS。

MANY 修饰符不能与 AND、OR、ACCURIE 或关系运算符一起使用。

NOT 修饰符

将某个单词或短语与 NOT 修饰符一起使用可排除出现该单词或短语的文档。第 93 页的“主题大纲示例”中的示例说明了如何使用 NOT 修饰符以检索提到“bebop”但未提到“fusion”的文档。

ORDER 修饰符

使用 ORDER 修饰符可指示搜索元素必须以何种顺序出现。如果搜索值没有按指定顺序出现在文档中，则不选取该文档。始终将 ORDER 修饰符放在紧靠运算符前面的位置。

只能将 ORDER 修饰符与 ALL、PARAGRAPH、SENTENCE 和 NEAR/N 运算符一起使用。

权值和文档重要性

本节说明为主题中的搜索条件赋予权值，以及权值对选取的文档的影响。涉及的具体信息包括：

- 哪些运算符接受权值
- 权值对重要性的影响
- 分配权值
- 主题得分和文档重要性

主题权值

在处理搜索代理时，Verity 搜索引擎将在后台为每个选取的文档计算一个得分。文档得分可以在 1.0 到 0.01 的范围内。文档的得分越高，其相关性就越高。通过为搜索代理选取的文档分配得分，Verity 应用程序可以向应用程序用户显示按相关性降序排列的结果。

文档的排列由组成搜索条件的元素决定。搜索条件是否包含主题以及主题是否包含权值将会影响到文档的排列。

在创建主题时，可以为主题结构分配权值，以指出主题定义的特定方面的相对重要性。例如，您可能对两个相关的对象感兴趣，但其中一个对象比另一个更重要。在编写主题时不必分配权值，因为当对主题集合建立索引时，会适当分配缺省权值。但是，通过分配权值可以调节所查找的内容的重要性。

哪些运算符接受权值

权值与运算符结合使用可以在搜索期间为父主题和子主题计算得分。为子主题指定的权值可以是介于 0.01 和 1.00 之间的数。一个子主题的权值指出了它相对于已为其父主题定义的其它子主题的重要性。子主题的权值越高，则认为其相对于其父主题的其它子主题的重要性越高。

只能将权值分配给使用概念运算符的主题的子主题，这些运算符包括：

- AND
- OR
- ACCRUE

不能为使用邻近运算符 SENTENCE 和 PARAGRAPH 的主题分配权值。这些运算符只简单假定其子项的“是”或“否”存在性。

如果被分配了邻近运算符的主题反过来又是已分配了概念运算符（如 AND 运算符）的主题的子主题，则可以为该子主题分配权值。

为一个主题的子主题分配权值并非强制性的，而仅仅是因为该运算符能够接受加权后的子主题。未分配权值时，子主题具有一个根据其运算符自动分配的权值。使用 AND 和 OR 运算符的主题的子项拥有权值 1.00，使用 ACCRUE 运算符的主题的子项有权值 0.50。如果更改了这些运算符，例如将 OR 运算符改变为 ACCRUE 运算符，那么还未具体赋予权值的那些子项的权值将相应更改。因此，如果一个 AND 主题的未分配权值的子项拥有权值 1.00，则当该运算符更改为 ACCRUE 时，此权值将变为 0.50。

如果为一个主题子项分配可变权值，那么请将随父项使用的运算符更改为不接受加权子项的运算符，例如 SENTENCE 运算符。当此运算符生效时，Verity 搜索引擎将自动假设权值为 1.00。如果该运算符随后更改为接受可变加权子项的运算符，那么以前分配的可变权值将再次变为有效。

权值对重要性的影响

为使用概念运算符的主题的子项分配权值时，就指定了该子项对由某主题产生的整体得分的相对贡献。为该子项分配的权值越高，选取的包含该子项的文档在结果列表中的出现位置就越高。因此，权值直接影响所选文档的重要性或排序。

例如，假定您拥有如下主题：



实证主题 80286 和 80386（它描述了 PC 产品中使用的微处理器）拥有自动赋予的权值 1.00。实证主题 486、386 和 286 具有引用其父主题的相对较高的概率，因此这些实证主题被赋予权值 0.80。实证主题 clone（圆锥）可能引用或根本不引用 PC clone，因此该实证主题被赋予权值 0.40。

使用此主题及其权值的搜索代理可能为匹配的文档产生下列得分：

Scores

1.00	01-Oct-90	New Toshiba Portable Desktop Computers Offer a Serious Alternative for Desktop Users
1.00	14-Feb-90	Technology: 'Chip Set' Unveiled for Use in Making Faster Computers
0.80	13-Feb-91	CMS Enhancements Inc. Unveils New Products
0.80	01-Oct-90	Top Selling Microsoft Windows Applications Now Available in One Convenient Package
0.80	01-Oct-90	KeyCorp Successful Bidder to Acquire New York State Divisions of Empire of America Federal
0.80	15-Feb-90	Health: U.S. Birth Control Lags

如果更改了每个实证主题的权值，则所选取的结果的重要性也会受影响。在此示例中，如果将实证主题 486 的权值更改为 0.60，将实证主题 386 的权值更改为 0.45，将实证主题 286 的权值更改为 0.35，将实证主题 clone 的权值更改为 0.20，则所选文档的得分将相应地更改，如下所示：

```

Scores
|
1.00 01-Oct-90 New Toshiba Portable Desktop Computers Offer a Serious Alternative for Desktop Users
1.00 14-Feb-90 Technology: 'Chip Set' Unveiled for Use in Making Faster Computers
0.60 13-Feb-91 CMS Enhancements Inc. Unveils New Products
0.60 01-Oct-90 KeyCorp Successful Bidder to Acquire New York State Divisions of Empire of America Federal
0.60 15-Feb-90 Health: U.S. Birth Control Lags
0.45 01-Oct-90 Top Selling Microsoft Windows Applications Now Available in One Convenient Package

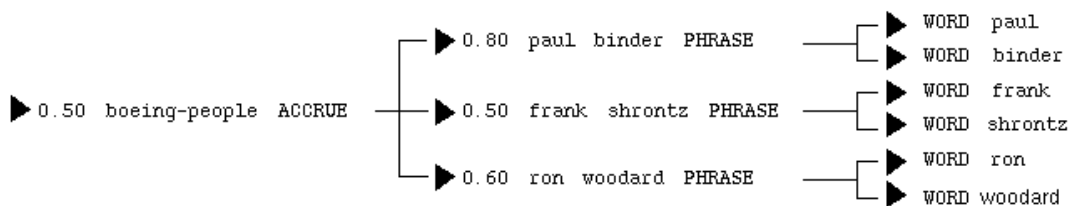
```

分配权值

为某个子项分配权值时，切记所使用的权值反映了一个子项对其父主题的重要性。匹配的文档将按对搜索的重要性进行排列；因此，所分配的权值将直接影响选取结果。如果更改权值，选取结果也将相应地更改。

示例：

主题 boeing-people 包括三个加权子项 binder、shrontz 和 woodard，如下所示。



这些子主题被赋予下列各种权值：子主题 binder 赋予权值 0.80，因为此子主题被认为是三个子主题中最重要。子主题 hitsman 赋予“中等”权值 0.50，因为该子主题相对于其它两个子主题是较为重要的。子主题 johnson 被分配低权值 0.30，因为该子主题被认为相对其它子主题而言最不重要。

当使用主题 `boeing-people` 进行搜索时，Verity 搜索引擎假定如果短语“Paul Binder”位于一个文档内，则该文档将与使用主题 `boeing-people` 的搜索相关的概率较高。包含短语“Frank Shrontz”的文档将与该搜索中等程度相关；包含短语“Ron Woodard”的文档与该搜索的相关性最小。

由于主题 `boeing-people` 已被分配 ACCRUE 运算符，因此，显示在结果列表顶部的文档将是那些包含最多子主题的文档；因此引用全部三个人的所有文档将具有最高的重要性。对于仅包含一个姓名的文档，将按照反映每个子项的权值的顺序进行选取。因此，由于 `binder` 子项具有最高权值，故只包含单个人名的文档将按照下列顺序进行排列：首先是引用 Paul Binder 的文档，其次是引用 Frank Shrontz 的文档，最后是引用 Ron Woodard 的文档。

自动分配权值

在创建子主题时，Verity 搜索引擎自动为使用 ACCRUE 运算符的主题的子项分配缺省权值 0.50。为使用 AND 或 OR 运算符的主题的子项自动分配权值 1.00。这些缺省权值可手工调大或调小，如第 109 页的“更改权值”所述。当基于使用邻近运算符的主题创建实证主题时，将为其分配缺省权值 1.00，且不能更改此权值。

分配权值的技巧

在初次分配权值时，开始为 ACCRUE 主题的子项分配权值 0.50，为其它所有主题的子项分配权值 1.00。

当为使用 ACCRUE 运算符的主题的子项分配权值时，如果子项没有过高的权值，则可以选择更为相关的结果。例如，为一个 ACCRUE 主题的所有子项分配权值 1.00 时，将使所有文档具有同等的重要性，而不管文档内出现多少个子项。对于只包含一个子项的所有文档以及包含所有子项的文档，Verity 搜索引擎将分配同等的重要性，因此在查看选取结果时，将无法区分这些文档。

为最佳的选取结果分配介于 0.80 和 0.20 之间的权值。

更改权值

在为子项分配权值后，可以通过使用父主题运行一个搜索来查看是否选取了所需的文档，以测试这些权值。如果发现需要更改权值，可以编辑为该子主题或实证主题分配的现有权值。在编辑主题大纲文件中的主题定义时，必须使用 `mktopics` 重新建立主题集。有关使用 `mktopics` 的完整信息，请参见 Verity 应用程序的管理指南。

主题得分和文档重要性

在使用主题执行搜索时，搜索代理将从考虑该主题的实证主题开始分析。如果存在实证主题，则将其分配 1.00 分，并将其视为与搜索相关。如果不存在实证主题，则为其分配 0.00 分，并将其视为与搜索不相关。如果实证主题是加权的，则实证主题的得分将乘以权值，然后按父主题的运算符指定的方式综合所得乘积。如果此父主题反过来又是正在搜索的其它主题的子主题，其得分将乘以它已分配有的权值，然后按父主题的运算符指定的方式综合所得乘积和其同级主题的乘积。重复此过程，直至到达父主题为止。

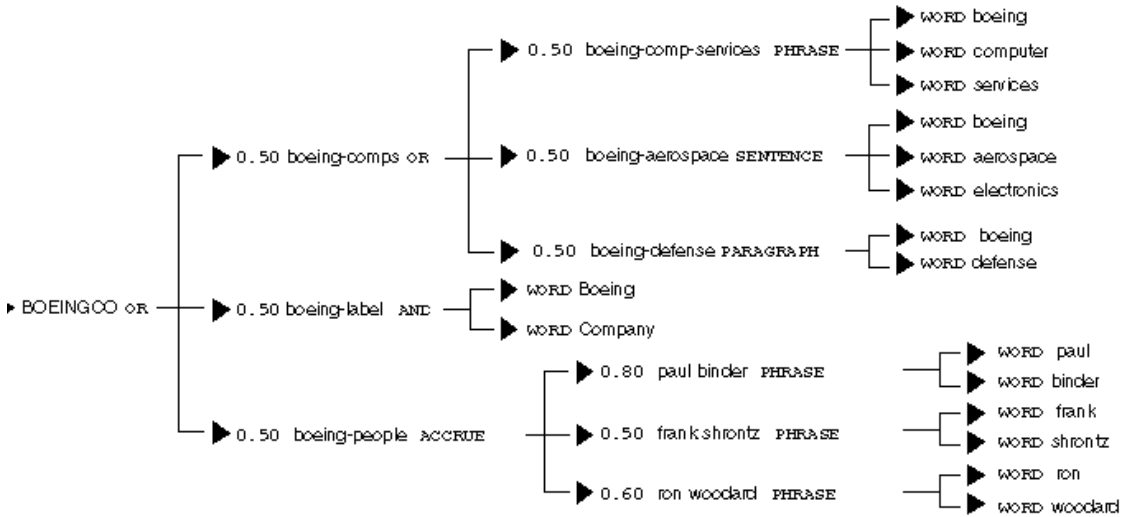
所使用的运算符确定了父主题和子主题的得分对选定文档的重要性的贡献。当为主题中的每个子项分配一个重要性分值时，将执行下列计算：

- 如果一个主题使用 `ACCRUE` 运算符，则最高排列结果将取自每个子主题的权值与得分之乘积，然后将文档中出现的每个子主题的得分增加一点。
- 如果一个主题使用 `AND` 运算符，则将对每个子主题自身的权值与得分之乘积进行比较，并将最小乘积（最小值）取作得分值。
- 如果一个子主题使用 `OR` 运算符，则将对每个子主题自身的权值和得分之乘积进行比较，并将最大的乘积（最大值）取作得分值。
- 如果一个子主题使用邻近运算符（`PHRASE`、`SENTENCE` 或 `PARAGRAPH`）或关系运算符，那么，如果主题存在，则该子主题将获得分值 1.00；如果主题不存在，则该子主题将获得分值 0.00。
- 一个实证主题如果存在则获得分值 1.00，如不存在则获得分值 0.00。

在针对父主题执行最终计算后，匹配的文档将可用于 Verity 应用程序，这样用户便可以查看其主要内容。

下面的示例提供了如何对实证主题和子主题进行计算的分类，以说明为选取的文档分配重要性的过程。

下面的示例则说明了在搜索中使用了父主题 BOEINGCO。



首先对照文档检查每个子主题的实证主题，以确定其是否存在。实证主题如果存在，则获得分值 1.00；如果不存在，则获得分值 0.00。

位于主题结构下一级的运算符将用于综合实证主题的分值。由于这一级的运算符都是邻近运算符（故不分配权值），因此它们都产生 0.00 或 1.00 的分值。

例如，假定下列实证主题出现在一个给定的文档内：

- 实证主题 “Boeing Computer Services” 出现在一个短语内。
- 实证主题 “Boeing Defense” 出现在一个段落内。实证主题 “Boeing Company” 出现在该文档内。
- 实证主题 “Ron Woodard” 出现在一个短语内。

其它实证主题仅部分存在或不存在。表 8-9 说明这些实证主题的存在性或不存在性如何影响主题得分。每个主题的得分都反映所有相关实证主题的存在性，这基于已分配给父主题的运算符。

表 8-9：实证主题和得分

主题	实证主题	实证主题存在	实证主题不存在	主题得分
boeing-comp-services	boeing computer services	1	1	
		1		
		1		

主题	实证主题	实证主题存在	实证主题不存在	主题得分
boeing-aerospace	boeing aerospace electronics	1	1 1	0
boeing-defense	boeing defense	1 1		1
boeing-label	boeing company	1 1		1
paul-binder	paul binder		1 1	0
frank-shrontz	frank shrontz		1 1	0
ron-woodard	ron woodard	1 1		1

在给定上面的主题得分后，对位于主题结构中下一层主题上的运算符进行下列计算：

- 子主题 boeing-comps（使用 AND 运算符）的得分为 0.50。
- 子主题 boeing-people（使用 ACCRUE 运算符）的得分为 0.50。

最后，主题 BOEINGCO（使用 OR 运算符）将比较每个子主题的权值与得分的乘积，并取最大的乘积（最大值）作为其得分。因此所选取的文档的得分为 0.50。

将对每个文档重复此过程。文档按 BOEINGCO 主题的得分进行排序，并按排列顺序显示。

设计主题

本节讨论可用于设计有效主题的方法。无论是准备用主题大纲文件还是某个 Verity 客户端程序编写主题，都可以应用此处介绍的方法和策略。本部分包括下列内容：

- 准备主题设计
- 主题设计策略
- 设计初始主题

准备主题设计

在准备主题设计时，应考虑将要使用的命名约定。主题名应当有助于识别要查找的文档的主题。

对主题名使用字母数字字符（A-Z，0-9）可保证最佳搜索性能。也可以使用 ASCII 值大于或等于 128 的外文字符，以及下面这些符号：\$（美元符）、%（百分号）、^（尖号）、+（加号）、-（减号）和_（下划线）。如果使用其它非字母数字字符，可能会造成对主题名的错误解释并影响结果。

了解信息需求

应了解主题设计所应用于的科目范围，并熟悉您站点上用户的搜索需求。下一步是了解信息需求以及要搜索的文档类型。

在计划初始主题设计时，请记住您是在制定一个策略，并且您定义的主题是将用来实施该策略的战术。

在制定策略时，请尝试回答下列问题：

- 您希望通过使用 Verity 搜索代理获得什么？
- 要由 Verity 搜索代理解决什么问题？
- 谁将使用搜索代理？
- 将使用哪些种类的源材料？
- 将执行哪些种类的搜索？
- 当前是如何执行搜索的？

将定义的主题视为要提出的问题。正如您可能在当地图书馆就相关的科目范围询问图书馆管理员一样，在建立 Verity 搜索代理时，您所创建的主题也应提出一些问题。

在考虑您的战略时，以及考虑 Verity 搜索应用程序将如何实现以提供答案时，要牢记您所设计的每个主题要扮演如下几个角色：

- 一个图书馆管理员
- 一个研究助理
- 一个信息存储库
- 一个知识库

了解文档

要建立有效的主题，必须较好地了解要用作信息源的文档类型。例如，您的文档可能包含下面的一类或多类信息：

- 信件
- 备忘录
- 报告
- 论文

收集要搜索的文档类型的代表性样本。注意将需要应用于所设计的主题的公共特征。例如，如果您的文档包含重要的术语、首字母缩写词或行话，可将其突出显示，以便创建能搜索这些文本的主题。

在收集文档样本时，标识其来源，无论它是内部来源（如内部审计报表）还是外部来源（如来自外部组织的电子邮件消息）。此信息使您可以为顶层主题定义子主题。

使用扫描数据

如果文档是使用 OCR 工具扫描成电子文件的，那么在建立索引之前，需确定是否应检查这些文件的准确性。如果扫描的文件已复审，应当向复审者咨询，确保对术语、首字母缩写词和行话应用了相关的标准。如果扫描文件未复审，应注意可能发生的变化。可以创建一个使用 OR 运算符的主题来包括各种变化形式。

文档样本分类

在收集代表性的文档样本并初次分析其内容后，可能需要对其进行进一步的分类。分类过程有助于定义主题设计中包含的顶层主题和子主题，并有助于确定要分配的运算符和权值。

下面是一些分类示例：

- 地理位置
- 适用人群
- 项目
- 科目范围
- 日期

分类过程有助于了解信息来源中存在的共同的、有意义的元素。例如，如果按日期（如一个月）对信息进行分类，那么创建使用 EQUALS 等关系运算符的主题就是有意义的。

主题设计策略

在了解了文档之后，就可以选择主题设计策略了。主题设计策略有两种：

- “自上而下”策略首先考虑主要主题的分类，然后再考虑逐步增加细节的分类。
- “自下而上”策略首先考虑细节领域，然后考虑用更广义的主题组合每个细节领域的分类。

自上而下设计

自上而下策略假定设计主题时从顶层主题开始，直到遍历每个子主题的各个实证主题。自上而下进行设计时，必须采用一个分类法或科学分类法按如下方式创建主题：

- 顶层主题 — 使用一般性标题标识科目范围
- 子主题 — 使用更具体的标题来标识科目范围内的主要分组，以及逐步变得更具体的主题。
- 实证主题 — 使用重要的术语、首字母缩写词或行话来定义主题对象。

当您有了定义清晰的要求时，自上而下设计策略最有效。如果可搜索文档集合经常扩充或变化，此方法也是理想之选。例如，您可以使用此策略定义信息源中可能尚不明显的主题对象。请记住，如果您发现很多新文档包含主题设计中未标识的信息，则始终可以添加新主题。

如果信息源（即索引文档集合）经常变化，则可能会遗漏文档内的特定主题对象，尤其是在最低层次上更容易如此。应定期分析按主题选取的信息，以确保对您的应用很关键的主题是最新的，并且能找到恰当的信息。

自下而上设计

自下而上策略在设计主题时，从各个实证主题开始向上设计，直至到达要定义的顶层主题。对于此策略而言，主题设计目标是选择包含类似于较低层主题的信息的文档。

在使用自下而上设计时，可以从包含要搜索的单词或短语的优秀代表样本的文档开始。然后按照逐步升级的类别对这些单词进行分组。

当您具有代表包含相似信息的许多其它文档的文档时，自下而上设计最有效。当信息源不会进行大幅更改或增加时，此方法也很有用。

基于特定文档内容的主题设计可能会遗漏其它文档中的相关科目范围。例如，如果在样本文档中使用了一个名称，而该名称在其它文档中发生了变化，则在搜索时可能会遗漏该新名称。

此外，使用自下而上策略还意味着您的主题设计与用于创建主题的特定文档集合比较相符。这些文档不可能代表信息源中包括的所有文档。应定期检查搜索的有效性。

设计初始主题

决定对初始主题设计使用自上而下方式还是自下而上方法后，需要创建一个主题大纲以标识要定义的主题层次。

制定主题大纲

制定主题大纲有助于确定如何在主题内的各个层次上对信息进行分类。可以对自上而下和自下而上设计方法使用主题大纲，不过主题大纲对自上而下方法尤其有效。建议先将建立的每个主题制作为一个大纲，以便理解主题与子主题之间的关系，并将其组织成最有用的形式。

主题大纲有助于您了解在您的站点上使用 Verity 搜索代理的人们如何搜索信息。可以使用主题大纲细微调整由主题和子主题指定的信息，直至精确选择文档为止。在制定主题大纲时尝试完成下列步骤：

- 识别人们在执行搜索时将使用的特定信息领域。
- 识别可以分组为一个父主题下的子项的所有相关子主题。
- 考虑主题设计要包括的初始细节级别。

开始时，使主题大纲保持在一个相对较小的范围。较小、较简单的主题大纲更容易定义，并且以后可以随时添加其它信息。在制定主题大纲时，确定主题设计将包括的层次数。

自上而下主题大纲示例

制定自上而下主题大纲包括三个步骤：

- 建立信息层次结构
- 建立各个搜索类别
- 建立要创建的主题

在执行这些步骤时，应当与在您的站点上使用 Verity 搜索代理的人们沟通一下，以制定出最能满足其搜索需求的主题大纲，如下所述。

建立信息层次结构

与这些目标用户进行交流，了解哪些类型的文档包含他们所需要的信息。

例如，假定您正在为医药行业的人们制定主题设计，以查找与当前药物测试相关的信息。通过与即将在您的站点上使用 Verity 搜索代理的人们进行讨论，您了解到下列类型的文档是当前药物测试信息的主要来源：

- 研究报告
- 产品说明书

这些文档构成了 Verity 搜索代理要搜索的信息来源。

建立各个搜索类别

检查将构成您的站点上的信息来源的文档，寻找对文档进行分类的方法。

在此示例中，通过查阅医学研究报告和产品说明书后得知，这些文档中的信息分成多个类别。确定下列类别在主题设计中将用于定义顶层主题：

- 实验报告
- 临床试验、数据或研究
- 产品说明书

确定要创建的主题

与在您的站点上创建 Verity 搜索代理的人们讨论您定义类别，确定选取的文档应包含的最重要的概念，并确定需要为每个类别制定的顶层主题。

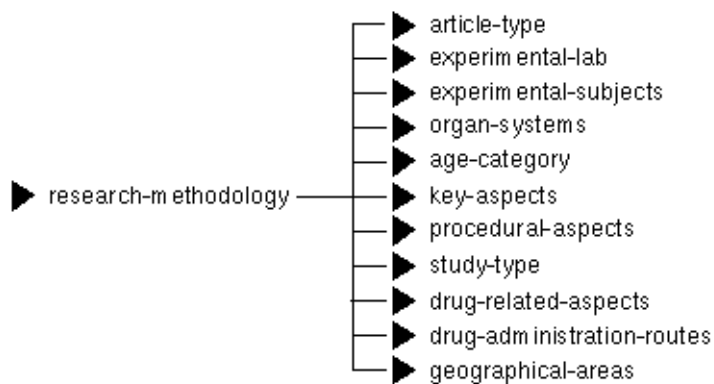
例如，您确定“临床试验”类别包含下列顶层主题：

product-testing

research-methodology

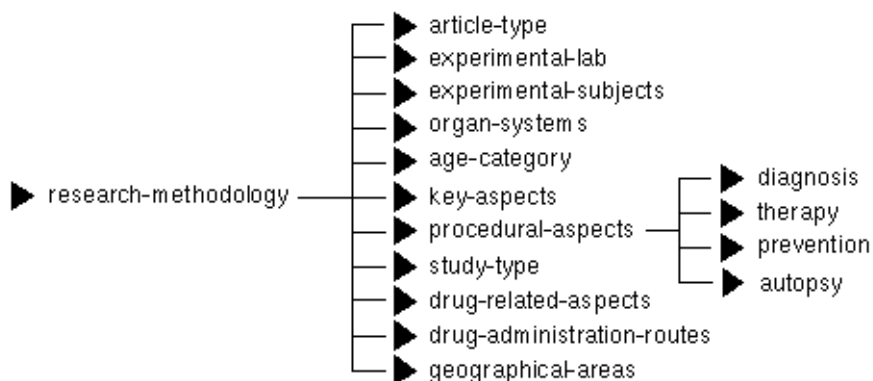
例如，在这些顶层主题内，由学科领域专家确定下列子主题：

▶ product-testing —▶ drug-names

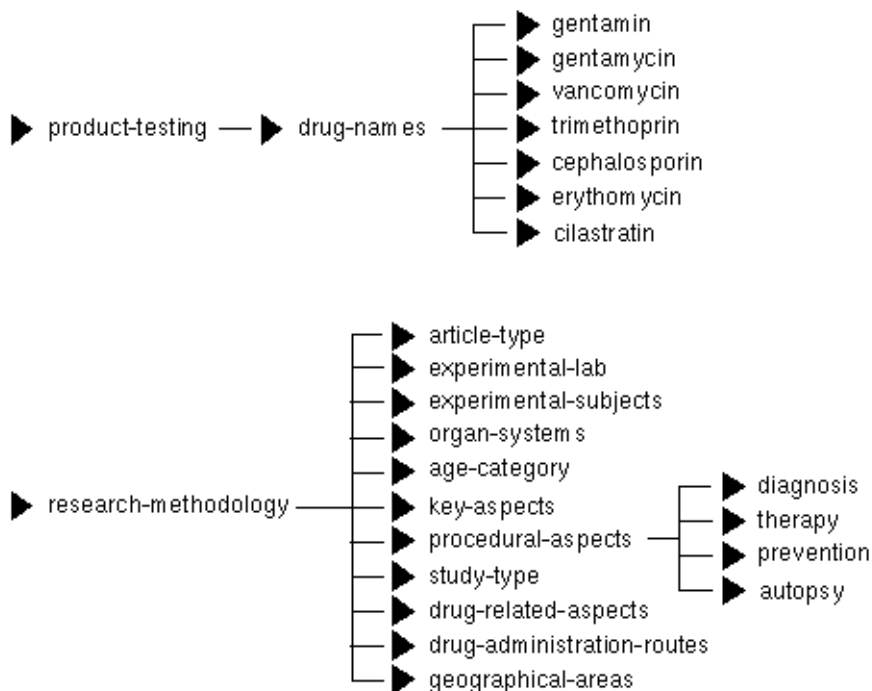


在对这些主题进行分类后，与在您的站点上使用 Verity 搜索代理的人们协商，以确定子主题。下面是在主题程序方面，将子主题分类为子项的示例：

▶ product-testing —▶ drug-names



在定义主题大纲后，与在您的站点上使用 Verity 搜索代理的人们协商，以确保这些主题选取有意义的文档。在下面的示例中，名为“drug-names”的主题允许您站点上的用户根据名称搜索药物的临床试验数据。



自下而上主题大纲示例

制定自下而上主题大纲包括三个步骤：

- 确定将构成主题设计的最低层的子主题
- 将相关子主题分类到更高层的主题中
- 确定顶层主题分类

在执行这些步骤时，应当与在您的站点上创建 Verity 搜索代理的人们沟通一下，以制定出最能满足您的搜索需求的主题大纲，如下所述。

标识低层主题

查找一个可用作样本的文档，其中所包含的信息代表了要查找的其它文档的信息。

例如，假定您正在制定一个主题设计来查找计算机行业方面的信息。首先创建一个主题，该主题搜索与 Apple Computer 和相关产品有关的文档。

使用下面的样本作为模型文档，其中所包含的信息代表了要查找的其它文档的信息：

A system developed specifically for networked Apple Computer, Inc. Macintosh computers has been announced by Human Designs, Inc.

Dubbed Chorus, the floor-standing unit reportedly can contain up to 16 floating-point processors and connects to networked Macintoshes to create a multiuser desktop environment.

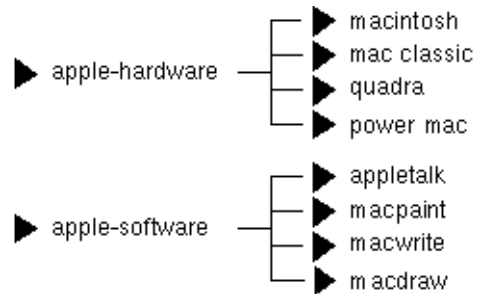
The product offers performance of eight million to 32 million floating-point operations per second and was designed to accommodate software development, according to the vendor. Options include an Ethernet I/O upgrade and a software simulator.

A Chorus 1 single floating-point processor entry-level system costs \$9,700. A Chorus 4 configuration with four floating point processors is available at \$25,000, which includes a dedicated I/O processor with an Apple Appletalk port and system software. Both systems are upgradable.

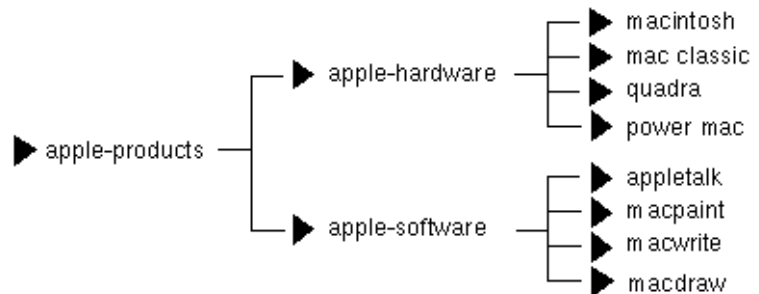
Human Designs, 322 W. 71st St., New York, N.Y. 10023. 212-580-0257.

此文档使您决定要查找引用“Appletalk”和“Macintosh”的其它文档，因此定义两个父主题名称：apple-software 和 apple-hardware。

您决定要添加其它实证主题，以选取包含相关信息的文档，如“Macintosh”、“Mac Classic”、“Quadra”和“Power Mac”。另外，您决定要包括实证主题“AppleTalk”、“MacPaint”、“MacWrite”和“MacDraw”作为相关的软件产品。按如下方式将这些实证主题分配给 apple-hardware 和 apple-software 主题：



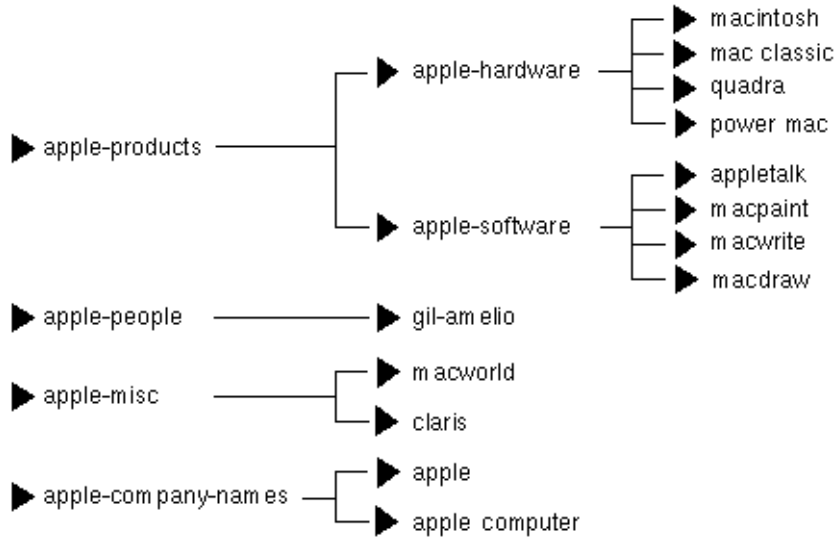
最后，需要将这些主题组合到 apple-products 主题中，如下所示：



对相关子主题进行分配

与将在您的站点上使用 Verity 搜索代理的人们讨论子主题，确定是否存在可在逻辑上归入一个类别的其它子主题。

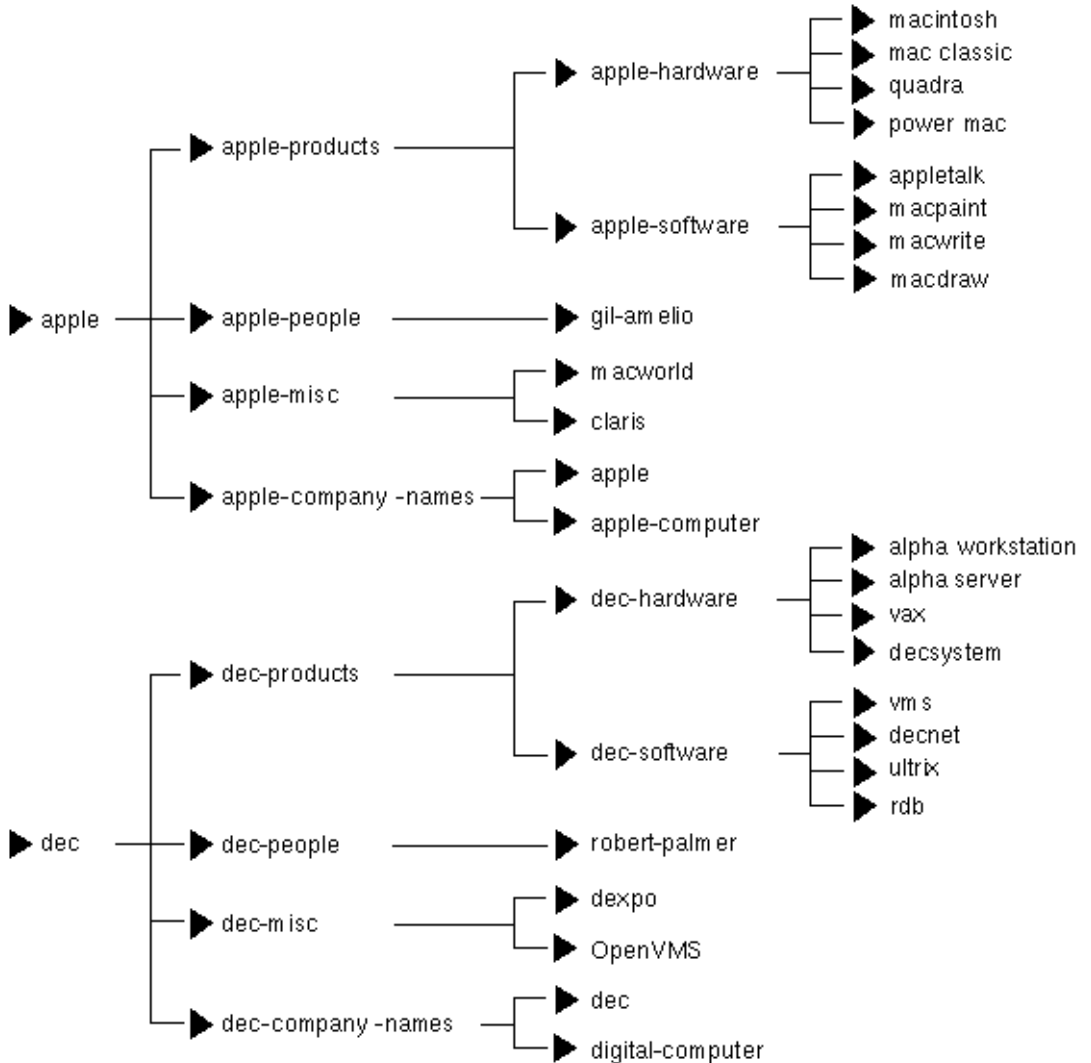
在此示例中，使用 Verity 搜索代理的部分人对查找 Apple Computer 公司的人事信息感兴趣，其他人则只想查找引用 Apple Computer 的文档。在下面的示例中，一个逻辑主题组可搜索 Apple Computer 的多方面信息：



建立顶层主题

确定其它顶层主题对于查找相关信息是否必要。

在下面的示例中，为另一个计算机公司 Digital Equipment Corporation 建立了一个新主题，即 dec。该主题被指定为顶层主题，并包含类似于为 apple 主题定义的子主题，如下所示。



Verity® 和 TOPIC® 都是 Verity, Inc. 的注册商标

系统过程

本附录描述 Sybase 提供的系统过程，这些过程用于从系统表获取报告和进行更新。表 A-1 列出了随增强型全文本搜索引擎提供的系统过程。

表 A-1: 系统过程

过程	说明
sp_check_text_index	报告或修复增强型全文本搜索索引和源表中的一致性问题。
sp_clean_text_events	从 <code>text_events</code> 表中删除已处理的条目。
sp_clean_text_indexes	删除不与表关联的文本索引。
sp_create_text_index	创建外部文本索引。
sp_drop_text_index	删除文本索引。
sp_help_text_index	显示文本索引。
sp_optimize_text_index	运行 Verity 优化例程。
sp_redo_text_events	更改 <code>text_events</code> 表中条目的状态，并强制对修改后的表重建索引。
sp_refresh_text_index	向 <code>text_events</code> 表中添加一个条目，以反映对应源表的更新。
sp_show_text_online	显示有关当前联机的数据库或索引的信息。
sp_text_cluster	显示或修改集群选项。
sp_text_configure	显示或修改增强型全文本搜索引擎配置参数。
sp_text_dump_database	创建数据库中的文本索引的一个备份副本，并可选择转储 <code>text_db</code> 和当前数据库。
sp_text_kill	终止特定文本索引的所有连接。
sp_text_load_index	从备份中恢复文本索引。
sp_text_notify	通知增强型全文本搜索引擎 <code>text_events</code> 表已修改。
sp_text_online	使数据库可供 Adaptive Server 使用。

sp_check_text_index

说明	报告或修复增强型全文本搜索索引和源表中的一致性问题。
语法	<code>sp_check_text_index server, "index_name", "id_column", "fixit"</code>
参数	<p><i>server</i> 是文本服务器的名称。</p> <p><i>index_name</i> 是文本服务器的名称。</p> <p><i>id_column</i> 是源标识或主键列名。</p> <p><i>fixit</i> 如果为 FALSE，则只报告问题。如果是 TRUE，修复问题但不报告。</p>
示例	列出含有列名 <code>text.i_text</code> 的服务器 <code>textsvr</code> 上的问题： <pre>sp_check_text_index "textsvr", "text.i_text", "id", "false"</pre>
用法	<ul style="list-style-type: none">在使用 <code>sp_check_text_index</code> 之前，必须发出 <code>sp_dboption "select into", true</code>此过程：<ul style="list-style-type: none">对于在索引中没有匹配条目的源表条目，产生一个 <code>sp_refresh_text_index insert</code> 操作。对于没有源表条目的索引表条目，产生一个 <code>sp_refresh_text_index delete</code> 操作。对于存在重复索引条目的每个额外条目，产生一个 <code>sp_refresh_text_index delete</code> 操作。若要确定重复索引，请从索引表中选取所有 ID 值，将其放入一个临时表中。如果结果集有多于 64K 个 ID 值，必须把 <code>batch_blocksize</code> 配置参数从其缺省值 0 更改为 65536，以启用按块读取所返回的 Verity 信息。如果没有做到这一点，增强型全文本搜索将试图一次读取所有 ID 值，并出现 Verity 错误 “-27” 而失败。
消息	无。
权限	任何用户都可以执行 <code>sp_check_text_index</code> 。

sp_clean_text_events

说明	从 <code>text_events</code> 表中删除已处理的条目。
语法	<code>sp_clean_text_events [up_to_date]</code>
参数	<code>up_to_date</code> 日期和时间，删除在此时间之后处理的所有条目。
示例	删除 1998 年 1 月 15 日下午 5:00 或之前输入的数据： <pre>sp_clean_text_events "01/15/98:17:00"</pre>
用法	<ul style="list-style-type: none">• 如果不指定 <code>up_to_date</code> 参数，那么所有日期小于或等于 <code>up_to_date</code> 且状态设为 “processed” 的条目都将被删除。• 如果省略 <code>up_to_date</code> 参数，那么状态设置为 “processed” 的所有条目都将被删除。• 只有在您备份了与文本索引关联的集合后，才会从 <code>text_events</code> 表中删除条目。• <code>sp_text_dump_database</code> 自动运行。
消息	无。
权限	任何用户都可以执行 <code>sp_clean_text_events</code> 。
另请参见	sp_text_dump_database

sp_clean_text_indexes

说明	从 <code>vesaux</code> 表中删除不与表关联的索引。
语法	<code>sp_clean_text_indexes</code>
参数	无。
示例	<code>sp_clean_text_indexes</code>
用法	<ul style="list-style-type: none">• 此过程从 <code>vesaux</code> 和 <code>vesauxcol</code> 表读取条目，确认源表和对应的索引表都存在。如果缺少其中之一，将删除索引。
消息	<ul style="list-style-type: none">• Fetch 操作导致错误。• 无法为 <code>index_name</code> 删除对象定义！
权限	任何用户都可以执行 <code>sp_clean_text_indexes</code> 。

sp_create_text_index

说明	创建一个文本索引。
语法	<pre>sp_create_text_index server_name, index_table_name, table_name, "batch", column_name [, column_name ...]</pre>
参数	<p><i>server_name</i> 是增强型全文本搜索引擎的名称。</p> <p><i>index_table_name</i> 是索引表的名称。 <i>index_table_name</i> 的形式为 [dbname.[owner.]]table, 其中:</p> <ul style="list-style-type: none"> • <i>dbname</i> 是包含索引表的数据库的名称。 • <i>owner</i> 是索引表的所有者名。 • <i>table</i> 是索引表名。 <p><i>table_name</i> 是包含要为其索引的文本的源表名。 <i>table_name</i> 的形式为 [dbname.[owner.]]table。</p> <p><i>batch</i> “batch” 运算符（必须括在引号中）指示增强型全文本搜索在将每个批处理发送到 VDK 后重新分配每个会话。</p> <p><i>column_name</i> 是由文本索引所索引的列的名称。</p>
示例	<p>在 blurbs 表的 copy 列上，创建一个文本索引和一个名为 i_blurbs 的索引表：</p> <pre>sp_create_text_index "blue", "i_blurbs", "blurbs", " ", "copy"</pre>
用法	<ul style="list-style-type: none"> • 在单个文本索引中，最多可以索引 16 列。 • 可以对以下数据类型的列编制索引：char、varchar、nchar、nvarchar、text、image、date、time、datetime、smalldatetime、int、smallint 和 tinyint。 • <i>option_string</i> 的内容不区分大小写。 • <i>option_string</i> 使用一个空字符串 (" ") 表示“无选项”。

- 为 *option_string* 指定一个“空”值，以创建将立即删除的文本索引。这样会创建 Verity 集合目录和样式文件，但不填充集合。例如，当为集群配置单个表时，将创建文本索引并立即删除它。在编辑 *style.prm* 文件后，重新创建文本索引。请参见第 27 页的“编辑单个 *style.prm* 文件”。
- `sp_create_text_index` 将条目写入 `vesaux` 表中，并通知增强型全文本搜索引擎创建文本索引。
- `sp_create_text_index` 的执行是同步的。在创建索引之前，执行此系统过程的 Adaptive Server 进程将保持阻塞状态。索引大量数据可能需要长达数小时的时间。
- 当在两列或更多列上创建文本索引时，文本索引中的每一列都放入一个单独的文档区域中。该区域的名称就是列名。区域可用于将搜索范围限定于一个特定的列。有关详细信息，请参见第 50 页的“in”。
- 在创建索引后不要重新命名。
- 不能从一个事务内部运行 `sp_create_text_index`。
- “*column_name*”不能是 NULL。
- 在表 “*table_name*” 中不存在列 “*column_name*”。
- 索引表映射失败 — 终止创建文本索引。
- 无效的文本索引名 — “*index_name*” 已存在。
- “*parameter*” 不在当前数据库中。
- 服务器名称 “*server_name*” 在 `syssservers` 中不存在。
- “*table_name*” 不存在。
- “*table_name*” 不是有效的对象名。
- 表 “*table_name*” 不具有标识列 — 终止创建文本索引。
- 文本索引创建失败。
- 用户 “*user_name*” 不是数据库中的有效用户。

消息

权限

任何用户都可以执行 `sp_create_text_index`。

sp_drop_text_index

说明	删除索引表和文本索引。
语法	<code>sp_drop_text_index "table_name.index_table_name" [, "table_name.index_table_name" ...]</code>
参数	<p><i>table_name</i> 是与要删除的文本索引相关联的表名。 <i>table_name</i> 的形式为 <code>[dbname.[owner.]]table</code>，其中：</p> <ul style="list-style-type: none"> • <code>dbname</code> 是包含表的数据库名。 • <code>owner</code> 是表的所有者名称。 • <code>table</code> 是表名。 <p><i>index_table_name</i> 是要删除的索引表和文本索引的名称。 <i>index_table_name</i> 的形式为 <code>[dbname.[owner.]]index</code>。</p>
示例	删除与 <code>blurbs</code> 表关联的索引表和文本索引： <pre>sp_drop_text_index "blurbs.i_blurbs"</pre>
用法	<ul style="list-style-type: none"> • 首先， <code>sp_drop_text_index</code> 向增强型全文本搜索引擎发出一个远程过程调用 (RPC) 以删除 Verity 集合。然后，它从 <code>vesaux</code> 和 <code>vesauxcol</code> 表中删除关联条目，删除索引表，并删除索引表对象定义。 • 在一个 <code>sp_drop_text_index</code> 请求中最多可指定 255 个索引。 • 如果未指定 <code>database</code> 和 <code>owner</code>，则使用当前所有者和数据库。
消息	<ul style="list-style-type: none"> • 不能从一个事务内部运行 <code>sp_drop_text_index</code>。 • 索引 “<i>index_name</i>” 不是文本索引。 • “<i>parameter_name</i>” 不是一个有效名称。 • 服务器名称 “<i>server_name</i>” 在 <code>sys.servers</code> 中不存在。 • 无法删除索引表 “<i>table_name</i>”。该表必须手工删除。 • 用户 “<i>user_name</i>” 不是数据库 “<i>database_name</i>” 中的有效用户。 • <code>vs_drop_index</code> 因代码 “<i>code_name</i>” 而失败。
权限	任何用户都可以执行 <code>sp_drop_text_index</code> 。

sp_help_text_index

说明	显示当前数据库的文本索引列表。
语法	sp_help_text_index [<i>index_table_name</i>]
参数	<i>index_table_name</i> 是要显示的文本索引的名称。
示例	示例 1 显示所有索引： <pre>sp_help_text_index</pre> 示例 2 显示关于文本索引 i_blurbs 的信息： <pre>sp_help_text_index "i_blurbs"</pre>
用法	<ul style="list-style-type: none">• sp_help_text_index 只能与 Enhanced Full-Text Search Specialty Data Store 一起使用。• 如果指定 <i>index_table_name</i> 参数，则显示关于该文本索引的信息。这些信息包括：文本索引的名称、索引的 Verity 集合的名称、源表的名称、IDENTITY 或主键列的名称，以及创建该索引的增强型全文搜索引擎的名称。• 如果省略 <i>index_table_name</i>，则显示当前数据库中的所有文本索引的列表。
消息	<ul style="list-style-type: none">• 在数据库 “<i>database_name</i>” 中未找到文本索引。• 文本索引 “<i>index_name</i>” 在数据库 “<i>database_name.</i>” 中不存在。• 对象必须在当前数据库中
权限	任何用户都可以执行 sp_help_text_index。

sp_optimize_text_index

说明	在文本索引上执行优化。
语法	<code>sp_optimize_text_index index_table_name</code>
参数	<p><i>index_table_name</i></p> <p>是要优化的文本索引名。 <i>index_table_name</i> 的形式为 <code>[dbname.[owner.]]table</code>，其中：</p> <ul style="list-style-type: none">• <code>dbname</code> 是包含索引表的数据库的名称。如果其存在的话，则需要有 <code>owner</code> 或一个占位符。• <code>owner</code> 是索引表的所有者名。• <code>table</code> 是索引表名。
示例	<p>优化文本索引 <code>i_blurbs</code> 可提高查询性能：</p> <pre>sp_optimize_text_index "i_blurbs"</pre>
用法	<ul style="list-style-type: none">• <code>sp_optimize_text_index</code> 只能与 Enhanced Full-Text Search Specialty Data Store 一起使用。• 此系统过程将使增强型全文搜索引擎通过 Verity 优化例程来运行指定的文本索引。• 对于在禁用 Verity 优化（启用跟踪标志 11）后更新的文本索引，<code>sp_optimize_text_index</code> 对其优化非常有用。• 若要启用 MaxClean 优化，请启用跟踪标志 30。此跟踪标志应仅在维护期间使用，因为它可能会消耗额外的时间并干扰正常的使用。MaxClean 是 Verity 的一项优化功能，可删除过时的集合文件。
消息	<ul style="list-style-type: none">• “<i>index_table_name</i>” 不在当前数据库中。• “<i>index_table_name</i>” 不存在。• 索引 “<i>index_table_name</i>” 不是文本索引。• 对于远程服务器 “<i>server_name</i>”，不支持此过程。
权限	任何用户都可以执行 <code>sp_optimize_text_index</code> 。
另请参见	第 73 页的“更新现有索引”

sp_redo_text_events

说明	更改 text_events 表中条目的状态，并强制重建修改后的列的索引。
语法	sp_redo_text_events [<i>from_date</i> [, <i>to_date</i>]]
参数	<i>from_date</i> 是要修改的条目的日期范围的起始日期和时间。 <i>to_date</i> 是要修改的条目的指定日期范围的结束日期和时间。
示例	对在 1998 年 1 月 5 日下午 5:00 与 1998 年 2 月 12 日上午 8:30 之间修改的列重建索引： <pre>sp_redo_text_events "01/05/98:17:00", "02/12/98:08:30"</pre>
用法	<ul style="list-style-type: none">• 将 text_events 表中当前状态为 “processed” 的所有条目的状态重新设置为 “unprocessed”。通知增强型全文本搜索引擎需要执行重建索引操作。• 对于在从一个备份中恢复 Verity 集合后同步文本索引很有用。此过程在 sp_text_load_index 期间自动运行。• 如果省略了 to_date，则介于 from_date 和当前日期之间的状态为 “processed” 的所有条目都将被重置为 “unprocessed”。• 如果省略了 from_date 和 to_date，则 text_events 表中所有状态为 “processed” 的条目都将被重置为 “un-processed”。
消息	<ul style="list-style-type: none">• 不能没有 from_date 而单独指定 to_date。• 您未指定完整的范围。
权限	任何用户都可以执行 sp_redo_text_events。

sp_refresh_text_index

说明	更改文本索引的源表数据时，记录 <code>text_events</code> 表中的修改。
语法	<code>sp_refresh_text_index table_name, column_name, rowid, mod_type</code>
参数	<p><i>table_name</i> 是正在更新的源表名。 <code>table_name</code> 的形式为 <code>[dbname.[owner.]]table</code>，其中：</p> <ul style="list-style-type: none"> • <code>dbname</code> 是包含表的数据库名。 • <code>owner</code> 是表的所有者名称。 • <code>table</code> 是表名。 <p><i>column_name</i> 是要更新的列的名称。</p> <p><i>rowid</i> 是更改的行的 <code>IDENTITY</code> 或主键列值。</p> <p><i>mod_type</i> 指定更改的类型。必须是 <code>insert</code>、<code>update</code> 或 <code>delete</code> 之一。</p>
示例	在 <code>text_events</code> 表中记录您已更新了 <code>blurbs</code> 表的 <code>copy</code> 列。已更新的行的 <code>id</code> 为 2.000000:
用法	<pre>sp_refresh_text_index "blurbs", "copy", 2.000000, "update"</pre> <ul style="list-style-type: none"> • 用户维护文本索引的一致性。任何时候更新已索引的源数据时，都必须运行 <code>sp_refresh_text_index</code>，以使 <code>text_events</code> 表能反映出更改。这样就可保持集合与源数据同步。集合直到您运行 <code>sp_text_notify</code> 后才更新。 • 可以为非 <code>text</code> 和非 <code>image</code> 列创建发出 <code>sp_refresh_text_index</code> 命令的触发器。请参见第 20 页的“将更改传播到文本索引”。
消息	<ul style="list-style-type: none"> • 在表 “<code>table_name</code>” 中不存在列 “<code>column_name</code>”。 • 指定的 <code>mod_type</code> 无效 (“<code>mod_type</code>”)。正确的值是：INSERT、UPDATE 和 DELETE。 • 所有者 “<code>owner_name</code>” 不存在。 • 表 “<code>table_name</code>” 不存在。 • “<code>table_name</code>” 不是有效的名称。 • 未找到文本事件表。
权限	任何用户都可以执行 <code>sp_refresh_text_index</code> 。
另请参见	sp_text_notify

sp_show_text_online

说明	显示有关当前联机的数据库或文本索引的信息。
语法	<code>sp_show_text_online server_name [{INDEXES DATABASES}]</code>
参数	<p><i>server_name</i> 是请求将发送到的增强型全文本搜索引擎的名称。</p> <p>INDEXES DATABASES 指定该请求是否应包含有关联机索引或联机数据库的数据。缺省值为 INDEXES。</p>
示例	<p>示例 1 显示当前在 KRAZYKAT 增强型全文本搜索引擎中联机的所有索引：</p> <pre>exec sp_show_text_online KRAZYKAT</pre> <p>示例 2 显示当前在 KRAZYKAT 增强型全文本搜索引擎中联机的所有数据库：</p> <pre>exec sp_show_text_online KRAZYKAT, DATABASES</pre>
用法	<ul style="list-style-type: none">• <code>sp_show_text_online</code> 向增强型全文本搜索引擎发出一个远程过程调用，以检索有关当前联机的索引或数据库的信息。• 如果此过程的结果未列出数据库，可通过 <code>sp_text_online</code> 使所需的数据库联机。
消息	<ul style="list-style-type: none">• <code>sp_show_text_online</code> 对服务器 <i>server_name</i> 失败。• 参数值 “<i>value</i>” 无效。• 发送到服务器的 RPC 返回一个失败返回码。• 第二个参数必须是 INDEXES 或 DATABASES。
权限	任何用户都可以执行 <code>sp_show_text_online</code> 。
另请参见	sp_text_online

sp_text_cluster

说明	显示或更改活动线程的集群参数。
语法	<code>sp_text_cluster server_name, cluster_parameter [, cluster_value]</code>
参数	<p><i>server_name</i> 是增强型全文本搜索引擎的名称。</p> <p><i>cluster_parameter</i> 是集群参数的名称。表 A-2 列出了有关的值。</p> <p><i>cluster_value</i> 是分配给活动线程的集群参数的值。表 A-2 列出了有关的值。</p>

表 A-2: 集群配置参数

<i>cluster_parameter</i> 的值	<i>cluster_value</i> 的值
cluster_style	指定要使用的集群类型。有效值为： <ul style="list-style-type: none"> • fixed — 产生固定数目的集群。数目由 cluster_max 参数设置。 • coarse — 根据较少的粗集群自动确定要产生的集群数目。 • medium — 根据中等大小的集群自动确定要产生的集群数目。 • fine — 根据较小较细的集群自动确定要产生的集群数目。
cluster_max	指定当 cluster_style 设置为 fixed 时产生的集群的最大数目。值 0 表示由搜索引擎来确定要产生的集群的数目。
cluster_effort	指定搜索引擎应花费在查找一个好的集群上的时间。有效值为： <ul style="list-style-type: none"> • effort_default — 搜索引擎使用时间的缺省量。如果将 Verity 术语 “default” 用双引号 (“ ”) 引起来，则还可以使用该术语。 • high — 搜索引擎花费最长时间。 • medium — 搜索引擎花费较少的时间。 • low — 搜索引擎花费最少的时间。
cluster_order	指定返回集群内的行的顺序。有效值为： <ul style="list-style-type: none"> • "0" — 表示行按与集群中心相似的顺序返回。这表示为集群返回的第一行是集群中行的最典型代表。 • "1" — 表示行按照为集群提交它们时的相对顺序返回。例如，如果集群 1 包括查询找到的第一、第三和第七行，这些行将按它们在集群中的这种相对顺序返回。

示例

示例 1 将活动线程的 *cluster_order* 参数更改为 1:

```
sp_text_cluster KRAZYKAT, cluster_order, "1"
```

示例 2 显示 *cluster_style* 参数的当前值:

```
sp_text_cluster KRAZYKAT, cluster_style
```

用法

- Verity 集群算法尝试根据集群参数的值，将类似的行组合到一起。
- 如果指定了 *cluster_parameter* 参数，但省略了 *cluster_value* 参数，则 `sp_text_cluster` 显示所指定的集群参数的值。
- `sp_text_cluster` 不修改集群配置参数的值。*cluster_value* 仅对当前正在执行的线程有效。若要修改缺省值，请使用 `sp_text_configure`。
- 有关如何请求集群结果集的方法，请参见第 45 页的“使用伪列请求集群结果集”。

消息

- 对于远程服务器 “*server_name*”，不支持此过程。
- 参数值 “*value*” 无效。
- `sp_text_cluster` 失败（状态 = *status*）。

权限

任何用户都可以执行 `sp_text_cluster`。

另请参见

[sp_text_configure](#)

sp_text_configure

说明	显示或更改增强型全文本搜索引擎配置参数。
语法	<code>sp_text_configure server_name [, config_name [, config_value]]</code>
参数	<p><i>server_name</i> 是增强型全文本搜索引擎的名称。</p> <p><i>config_name</i> 是要显示或修改的配置参数的名称。</p> <p><i>config_value</i> 是为配置参数指定的值。</p>
示例	<p>示例 1 将备份目标目录更改为 <code>/data/backup</code>:</p> <pre>sp_text_configure KRAZYCAT, backdir, "/data/backup"</pre> <p>示例 2 显示备份目标目录:</p> <pre>sp_text_configure KRAZYCAT, backdir</pre>
用法	<ul style="list-style-type: none"> 当执行 <code>sp_text_configure</code> 以修改动态参数时: <ul style="list-style-type: none"> 将更新配置值和运行值。 将更新配置文件。 更改会立即生效。 当执行 <code>sp_text_configure</code> 以修改静态参数时: <ul style="list-style-type: none"> 将更新配置值。 将更新配置文件。 只有当重新启动增强型全文本搜索引擎时更改才会生效。 当不带参数发布时, <code>sp_text_configure</code> 会显示增强型全文本搜索引擎的所有配置参数及其当前值的报告。 如果指定了 <i>config_name</i> 参数, 但省略了 <i>config_value</i> 参数, 则 <code>sp_text_configure</code> 将显示所指定的配置参数的报告。 有关各配置参数的信息, 请参见第 62 页的“修改配置参数”。
消息	<ul style="list-style-type: none"> 如果没有配置选项, 将无法指定配置值。 对于远程服务器 “<i>server_name</i>”, 不支持此过程。 <code>sp_text_configure</code> 失败 – 可能是无效的配置选项 “<i>config_name</i>”。
权限	任何用户都可以执行 <code>sp_text_configure</code> 。

sp_text_dump_database

说明 创建文本索引的备份副本。

语法 sp_text_dump_database *backupdbs* [, *current_to*] [, *current_with*] [, *current_stripe01* [, ... [, *current_stripe31*]]] [, *textdb_to*] [, *textdb_with*] [, *textdb_stripe01* [, ... [, *textdb_stripe31*]]]

参数

backupdbs

指定是否在备份文本索引前备份当前数据库和 `text_db` 数据库。有效值为：

- `CURRENT_DB_AND_INDEXES` — 指定在备份文本索引之前备份当前数据库。
- `CURRENT_DB_AND_CURRENT_INDEXES` — 指定在备份文本索引之前备份当前数据库，并且只转储与当前数据库相关联的索引。
- `TEXT_DB_AND_INDEXES` — 指定在备份文本索引之前备份 `text_db` 数据库。
- `INDEXES_AND_DATABASES` — 指定在备份文本索引之前备份当前数据库和 `text_db` 数据库。
- `ONLY_INDEXES` — 指定只备份文本索引。

current_to

是用于转储当前数据库的 `dump database` 命令的 `to` 子句。仅当为 *backupdbs* 参数指定 `CURRENT_DB_AND_INDEXES` 或 `INDEXES_AND_DATABASES` 值时才使用此选项。

current_with

是用于转储当前数据库的 `dump database` 命令的 `with` 子句。仅当为 *backupdbs* 参数指定 `CURRENT_DB_AND_INDEXES` 或 `INDEXES_AND_DATABASES` 值时才使用此选项。

current_stripe

是用于转储当前数据库的 `dump database` 命令的 `stripe` 子句。仅当为 *backupdbs* 参数指定 `CURRENT_DB_AND_INDEXES` 或 `INDEXES_AND_DATABASES` 值时才使用此选项。

textdb_to

是用于转储 `text_db` 数据库的 `dump database` 命令的 `to` 子句。仅当为 *backupdbs* 参数指定 `INDEXES_AND_DATABASES` 值时才使用此选项。仅当为 *backupdbs* 参数指定 `TEXT_DB_AND_INDEXES` 或 `INDEXES_AND_DATABASES` 值时才使用此选项。

textdb_with

是用于转储 *text_db* 数据库的 `dump database` 命令的 `with` 子句。仅当为 *backupdbs* 参数指定 `TEXT_DB_AND_INDEXES` 或 `INDEXES_AND_DATABASES` 值时才使用此选项。

textdb_stripe

是用于转储 *text_db* 数据库的 `dump database` 命令的 `stripe` 子句。仅当为 *backupdbs* 参数指定 `TEXT_DB_AND_INDEXES` 或 `INDEXES_AND_DATABASES` 值时才使用此选项。

示例

示例 1 只备份文本索引：

```
sp_text_dump_database ONLY_INDEXES
```

示例 2 在备份文本索引之前，将当前数据库转储到 `/data/db1backup`：

```
sp_text_dump_database CURRENT_DB_AND_INDEXES, "to '/data/db1backup'"
```

示例 3 在备份文本索引之前，将 *text_db* 数据库转储到 `/data/textdbbackup`：

```
sp_text_dump_database @backupdbs = "TEXT_DB_AND_INDEXES",
@textdb_to = "to '/data/textdbbackup'"
```

示例 4 在备份文本索引之前，将当前数据库转储到 `/data/db1backup`，将 *text_db* 数据库转储到 `/data/textdbbackup`：

```
sp_text_dump_database @backupdbs = "INDEXES_AND_DATABASES",
@current_to = "to '/data/db1backup'",
@textdb_to = "to '/data/textdbbackup'"
```

用法

- 增强型全文本搜索引擎将 *current_to*、*current_with* 和 *current_stripe01* 到 *current_stripe31* 的值并置到 `dump database currentdbname`，然后执行 `dump database` 命令。执行 `dump database` 命令的输出被发送到增强型全文本搜索错误日志中。
- 增强型全文本搜索引擎将 *textdb_to*、*textdb_with* 和 *textdb_stripe01* 到 *textdb_stripe31* 的值并置到字符串 “`dump database currentdbname`”，然后执行 `dump database` 命令。执行 `dump database` 命令的输出被发送到增强型全文本搜索错误日志中。
- 当备份完所有索引后，将删除 `text_events` 表中所有在当前数据库中状态为 “`processed`” 的条目。
- Verity 集合的备份文件存放在由配置参数 `backDir` 所指定的目录中。
- 有关自定义备份的信息，请参见配置参数 `backCmd` 的参考信息。
- 参数值 “*value*” 无效。
- 服务器名称 “*server*” 在 `syssservers` 中不存在。

消息

- 试图转储数据库 “*database_name*” 失败 — 使用 `dump database` 命令。
- 试图在服务器 “*server_name*” 上备份文本索引失败。
- 试图在数据库 “*database_name*” 中清理 `text_events` 失败 (`date = 'date'`)。
- 在转储数据库 “*database_name*” 时, 参数 “*parameter_name*” 是必需的。
- 转储数据库 “*database_name*” — 查看全文本搜索 SDS 错误日志中的状态。

权限

任何用户都可以执行 `sp_text_dump_database`。

另请参见

《参考手册》中的 `dump_database`。

sp_text_kill

说明	终止特定文本索引的所有连接。
语法	<code>sp_text_kill index_table_name</code>
参数	<p><i>index_table_name</i></p> <p>是文本索引的名称，从中将终止所有连接。<i>index_table_name</i> 的形式为 <code>[dbname.[owner.]]table</code>，其中：</p> <ul style="list-style-type: none">• <code>dbname</code> 是包含索引表的数据库的名称。如果其存在的话，则需要有 <code>owner</code> 或一个占位符。• <code>owner</code> 是索引表的所有者名。• <code>table</code> 是索引表名。
示例	终止文本索引 <code>i_blurbs</code> 的所有现有连接： <pre>sp_text_kill "i_blurbs"</pre>
用法	<ul style="list-style-type: none">• 此系统过程使增强型全文本搜索引擎终止指定索引的所有连接，但发起请求的连接除外。• 试图删除当前正在使用的文本索引将会失败。<code>sp_text_kill</code> 可用于终止所有现有连接，以成功删除索引。
消息	<ul style="list-style-type: none">• 索引 “<i>index_table_name</i>” 不是文本索引。• 对于远程服务器 “<i>server_name</i>”，不支持此过程。• “<i>index_table_name</i>” 不存在。• 只有系统管理员 (SA) 才能执行此过程。
权限	只有用户 “sa” 才能执行 <code>sp_text_kill</code> 。
另请参见	sp_drop_text_index

sp_text_load_index

说明 恢复文本索引备份。

语法 `sp_text_load_index`

参数 无。

示例 `sp_text_load_index`

恢复当前数据库中的所有文本索引。

用法

- 在完全恢复 `text_db` 数据库和当前数据库后，运行 `sp_text_load_index`。

- `sp_text_load_index` 从最新的备份中恢复 Verity 集合。然后增强型全文搜索引擎运行 `sp_redo_text_events` 和 `sp_text_notify`，以便重新应用 `text_events` 表中自备份索引以来的所有条目。

消息

- 服务器名称 “*server_name*” 在 `sys.servers` 中不存在。

- 无法为服务器 “*server_name*” 恢复文本索引。

- 对于远程服务器 “*server_name*”，不支持此过程。

- 服务器 “*server_name*” 的数据库 `database_name` 中 `text_events` 表的更新失败 — `text_events` 未前滚。

权限 任何用户都可以执行 `sp_text_load_index`。

另请参见 [sp_redo_text_events](#) ; [sp_text_notify](#)

sp_text_notify

说明	通知增强型全文本搜索引擎 <code>text_events</code> 表已修改。
语法	<code>sp_text_notify</code> [{true false}] [, <i>server_name</i>]
参数	true 使过程同步运行。 false 使过程异步运行。 <i>server_name</i> 是所通知的增强型全文本搜索引擎的名称。
示例	<pre>sp_text_notify true</pre>
用法	<ul style="list-style-type: none">在发出 <code>sp_refresh_text_index</code> 后必须运行 <code>sp_text_notify</code>，以通知增强型全文本搜索引擎源表已修改。如果不指定 <code>true</code> 或 <code>false</code>，<code>sp_text_notify</code> 将同步运行。如果未指定服务器名称，则将通知所有增强型全文本搜索引擎。
消息	<ul style="list-style-type: none">不能从一个事务内部运行 <code>sp_text_notify</code>。通知失败，服务器 = “<i>server_name</i>”。服务器名称 “<i>server_name</i>” 在 <code>syssservers</code> 中不存在。参数值 “<i>value</i>” 无效。
权限	任何用户都可以执行 <code>sp_text_notify</code> 。
另请参见	sp_refresh_text_index

sp_text_online

说明	使一个数据库可用于 Adaptive Server 的全文本搜索。
语法	<code>sp_text_online [server_name], [database_name]</code>
参数	<p><i>server_name</i> 是增强型全文本搜索引擎的名称。</p> <p><i>database_name</i> 是要联机的数据库的名称。</p>
示例	使 pubs2 数据库可用于使用增强型全文本搜索引擎的全文本搜索： <pre>sp_text_online @database_name = pubs2</pre>
用法	<ul style="list-style-type: none">• 如果未指定数据库，则为全文本搜索联机所有数据库。• 如果未指定服务器名，则将通知 vesaux 表中列出的所有增强型全文本搜索引擎。• 使用增强型全文本搜索引擎时，数据库会在 auto_online 配置参数设置为 1 时自动联机。
消息	<ul style="list-style-type: none">• 使用文本索引的所有数据库现在都已联机• 服务器 “<i>database_names</i>” 上包含文本索引的数据库现在已联机。• 服务器 <i>server_name</i> 现在已联机。• 服务器名称 “<i>server_name</i>” 在 syssservers 中不存在。• 参数值 “<i>value</i>” 无效。• 指定的数据库不存在• vs_online 对服务器 “<i>server_name</i>” 失败。
权限	任何用户都可以执行 sp_text_online。

示例文件

本附录包含以下内容:

- 缺省配置文件 (*textsvr.cfg*) 的文本
- *sample_text_main.sql* 示例脚本概述
- 增强型全文本搜索引擎提供的所有示例文件列表
- *getsend* 程序概述

缺省 *textsvr.cfg* 配置文件

```

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;  @(#)  File: textsvr.cfg  1.17  07/26/99
;
;  Full Text Search Specialty Data Store
;          Sample Configuration File
;
;  The installation procedure places this file in the
;  "SYBASE" directory.
;
;  Lines with a semi-colon in column 1 are comment lines.
;
;  Modification History:
;  -----
;  11-21-97      Create file for Full Text Search SDS
;  03-02-98      Add trace flags and config values for
;                  Enhanced Full Text Search SDS
;  05-26-99      remove references to sds/text
;  07-09-99      added batch block size
;  08-24-99      remove version string and correct copyright
;
;
;
;          copyright (c) 1997, 1999
;          Sybase, Inc. Emeryville, CA

```



```

;          DEFINITIONS OF TRACE FLAG AND SORT ORDER VALUES
;
; "traceflags" parameter, for text server
; Available "traceflags" values:1,2,3,4,5,6,7,8,9,10,11,12,13
;
; 1 trace connect/disconnect/attention events
; 2 trace language events
; 3 trace rpc events
; 4 trace cursor events
; 5 log error messages returned to the client
; 6 trace information about indexes
; 7 trace senddone packets
; 8 write text server/Verity api interface records to the log
; 9 trace sql parser
; 10 trace Verity processing
; 11 disable Verity collection optimization
; 12 disable returning of sp_statistics information
; 13 trace backup operations (Enhanced Full Text Search only)
;
; "srv_traceflags" parameter, for Open Server component of text server
; Available "srv_traceflags" values:1,2,3,4,5,6,7,8
; 1 trace TDS headers
; 2 trace TDS data
; 3 trace attention events
; 4 trace message queues
; 5 trace TDS tokens
; 6 trace open server events
; 7 trace deferred event queue
; 8 trace network requests
;
; "sort_order" parameter
; Available "sort_order" values:0,1,2,3
; 0 order by score, descending (default)
; 1 order by score, ascending
; 2 order by timestamp, descending
; 3 order by timestamp, ascending
;
; ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
;
;          CONFIG VALUES SECTION
;
; The "textsvr.cfg" file is supplied with the values commented out.
; To override value(s) in the executable program:
;   - Set required value(s) below
;   - Remove the semicolon from column 1
;

```

```
[textsvr]
;min_sessions = 10
;max_sessions = 100
;batch_size = 500
;sort_order = 0
;defaultDb = text_db
;errorLog = textsvr.log
;language = english
;charset = iso_1
;vdkLanguage =
;vdkCharset = 850
;traceflags = 0
;srv_traceflags = 0
;max_indexes = 126
;max_packet_size = 2048
;max_stack_size = 34816
;max_threads = 50
;collDir = <textsvr directory tree location on UNIX>/collections
;collDir = <textsvr directory tree location on Win-NT>\collections
;vdkHome = <textsvr directory tree location on UNIX>/verity
;vdkHome = <textsvr location on Win-NT>\verity
;interfaces = <${SYBASE} location on UNIX>/interfaces
;interfaces = <%SYBASE% location on Win-NT>\ini\sql.ini
;::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
;
; The parameters in this section apply only to the Enhanced Full Text Search SDS.
; If defined to a Full Text Search engine they will be ignored.
;
;auto_online = 0
;backDir = <textsvr directory tree location on UNIX>/backup
;backDir = <textsvr directory tree location on Win-NT>\backup
;backCmd =
;restoreCmd =
;knowledge_base =
;nocase = 0
;cluster_max = 0
;cluster_order = 0
;cluster_style = Fixed
;cluster_effort = Default
;batch_blocksize = 0
;max_session_fd = 0
```


sample_text_main.sql 脚本

增强型全文本搜索引擎安装程序会将 `sample_text_main.sql` 脚本复制到 `$SYBASE/$SYBASE_FTS/sample/scripts` 目录。该脚本演示了以下操作：

- 设置文本索引。
- 修改数据并将更改传播到集合。修改包括插入、更新和删除。
- 删除文本索引。

安装或配置过程中不是必须执行该脚本； Sybase 提供该脚本作为示例。

在运行 `sample_text_main.sql` 脚本之前：

- 必须配置和运行 Adaptive Server 和增强型全文本搜索引擎。
- 使用文本编辑器编辑 `sample_text_main.sql` 脚本。将 `sample_text_main.sql` 脚本的步骤 4 中的 “YOUR_TEXT_SERVER” 更改为增强型全文本搜索引擎的名称。
- 确认 model 数据库包含 `text_events` 表。如果 model 数据库不是这样配置的，则必须：
 - 将 `sample_text_main.sql` 脚本修改为创建该数据库后退出
 - 对新数据库应用 `installevent` 脚本。请参见第 14 页的“运行 [installevent 脚本](#)”。
 - 执行示例脚本的余下部分

将该脚本指示为 Adaptive Server 的输入。例如，要在名为 MYSVR 的 Adaptive Server 上运行 `sample_text_main.sql` 脚本：

```
isql -Ulogin -Ppassword -SMYSVR
-i
$SYBASE/$SYBASE_FTS/sample/scripts/sample_text_main.sql
l -omain.out
```

配置完此示例环境后，登录到 Adaptive Server 并删除示例数据库。例如：

```
1> use master
2> go
1> drop database sample_colors_db
2> go
```

如果需要，可以重新运行 `sample_text_main.sql`。

演示增强型全文本搜索引擎功能的示例文件

增强型全文本搜索引擎提供一组用于演示文本服务器操作的示例文件。这些示例文件位于 `$$SYBASE/$SYBASE_FTS/sample/scripts` 目录中。安装、配置或操作增强型全文本搜索引擎的过程中不是必须执行这些示例文件。

自定义同义词库

以下文件演示了如何设置和使用自定义同义词库：

- `sample_text_thesaurus.ctl` — 是示例控制文件。
- `sample_text_thesaurus.sql` — 使用示例控制文件创建的自定义同义词库提供示例查询。

主题

以下文件演示了如何设置和使用主题：

- `sample_text_topics.otl` — 是示例大纲文件。
- `sample_text_topics.kbm` — 是示例知识库映射。
- `sample_text_topics.sql` — 使用定义的主题提供示例查询。

集群、汇总和按示例查询

以下文件演示了如何设置和使用集群、汇总和按示例查询：

- `sample_text_setup.sql` — 创建示例环境。
- `sample_text_queries.sql` — 向该环境发出查询并删除该环境。

getsend 示例程序

增强型全文本搜索引擎提供了一个名为 `getsend` 的程序，可将 `text` 或 `image` 数据从一个文件装载到 Adaptive Server 中定义的一个列。

建立和运行该程序所需的源文件和头文件、`makefile` 和目录包含在 `$$SYBASE/$SYBASE_FTS/sample/source` 目录中。

有关如何使用该程序的信息，请参见 `README.TXT` 文件和 `getsend.c` 文件。

Unicode 支持

Unicode 标准是国际标准化组织的 ISO 10646 标准的一个子集，是一个国际字符集。Unicode 完全等同于 ISO 10646 的基本多语言平面 (Basic Multilingual Plane, BMP)，可以支持世界上所有主要脚本和语言。因此，Unicode 是现有的所有字符集的一个超集。

Unicode:

- 提供单一源代码开发。这意味着您一次性开发一个应用程序之后，就可以针对多个地区和多种语言对其进行本地化。通过使用单个统一字符集，您不必修改应用程序来考虑字符集之间的差异，从而降低了开发、测试成本和技术支持成本。
- 允许在同一个数据库中混用不同的语言。一个纯 Unicode 系统不要求对数据库进行设计来跟踪数据的字符集。

增强型全文本搜索引擎支持 Unicode。若要使用此功能，请将 `charset` 配置值设置为 `utf8`。这个工具包中包含了安装支持 Unicode 的客户端 / 服务器数据库系统所需的一切内容。

若要配置增强型全文本搜索引擎以使用 Unicode 格式来存储数据，请将 `charset` 配置值设置为 `utf8`。请参见第 62 页的“修改配置参数”。

注释 如果您对 Unicode 格式的数据发出通配符搜索，请打开跟踪标志 15。请参见第 68 页的“设置跟踪标志”。

使用 XML 数据

本附录包含以下内容：

- 在字段和区域中使用 XML 数据的正确格式
- 通过字段和区域将 XML 数据编制到文本索引中的示例

针对字段和区域正确设置 XML 数据的格式

可以将 XML 数据放入 Adaptive Server 数据库中的文本列内，然后在数据上创建文本索引。这需要使用带有特殊语法（其中包括 Verity 运算符）的 `select` 语句。

例如，在下面的 `select` 语句中，`<in>` 是 Verity 运算符。

```
select t1.id
from
    ti_address_tbl t1,
    address_tbl t2
where
    t1.id=t2.id and
    t1.index_any ='USA <in> address'
```

Sybase 支持在 XML 文档组织良好且具有以下格式时，通过字段或区域将 XML 数据编制到文本索引中：

```
<address>
<street>123 Main St.</street>
<city>Anywhere</city>
<state>CA</state>
<country>USA</country>
</address>
```

此格式会将文本“123 Main St. Anywhere CA USA”插入地址区域，将“123 Main St.”插入街道区域，等等。

XML 索引示例

本节提供使用“[针对字段和区域正确设置 XML 数据的格式](#)”中的格式的示例。此示例可实现以下步骤：

- 1 在空表上创建空文本索引。
- 2 删除文本索引。此步骤会删除与 Adaptive Server 中的文本索引关联的代理表，但会保留 collections 目录结构中的所有缺省目录和文件以供 EFTS 文本索引使用。
- 3 修改 style.dft 文件以设置通用过滤器。
- 4 将数据插入表中。
- 5 重新创建文本索引。
- 6 发出 select 请求，以便使用文本索引根据 XML 数据来分离行。

示例片段

片段 1

此片段演示如何在空表上创建空文本索引。

```
isql -Sase1501sunbox -Usa -P -w2048 Dxml_db1
create table address_tbl
(
    id numeric(5,0)
    identity, xmlcol text
)
go
create unique index uidx on address_tbl(id)
go
sp_create_text_index 'textsvr', 'ti_address_tbl',
'address_tbl', "empty ", 'xmlcol'
go
(return status = 0)
quit
```

片段 2

此片段演示如何删除空文本索引。此步骤会删除与 Adaptive Server 中的文本索引关联的代理表，但会保留 collections 目录结构中的所有缺省目录和文件以供 EFTS 文本索引使用。

```
isql -Sase1501sunbox -Usa -P -w2048 Dxml_db1
sp_drop_text_index 'address_tbl.ti_address_tbl'
go
(return status = 0)
quit
```

片段 3

此片段演示如何修改 style.dft 文件来设置通用过滤器。

```
/sy/ase1501sunbox/EFTS-15_0/collections
    xml_db.dbo.ti_address_tbl/style
(uid=syuid)_sunbox> pwd
/sy/ase1501sunbox/EFTS-15_0/collections
    xml_db.dbo.ti_address_tbl/style
(uid=syuid)_sunbox>ls -al
Total 74
drwxr-xr-x  2 sybase sybase
512 Jan  8 16:14 .
drwxr-xr-x 11 sybase sybase
512 Jan  8 16:14 ..
...
-rw-r--r--  1 sybase sybase
288 Jan  8 16:14 style.dft
...
(uid=syuid)_sunbox> mv ./style.dft
./style.dft.orig
(uid=syuid)_sunbox> cp
./style.dft.orig ./style.dft
(uid=syuid)_sunbox> vi ./style.dft
(uid=syuid)_sunbox> diff
./style.dft.orig ./style.dft
11a12
> /filter="universal"
(uid=syuid)_sunbox>
```

片段 4

此片段演示如何将 XML 数据插入表中。

```
isql -Sase1501sunbox -Usa -P -w2048 -Dxml_db
insert address_tbl values
('<?xml version="1.0"
encoding="UTF-8"?>
<fulldoc>
<address>
<street>mailstop
101</street>
<city>CONCORD</city>
<state>MA</state>
<country>USA</country>
</address>
<filepath>
/sy/ase1501sunbox/EFTS-15_0/collections
/made_using_mkvdv
/xml_data
/row1_well_formed.xml
</filepath>
</fulldoc>
')
go
(1 row affected)
insert address_tbl values
('<?xml version="1.0"
encoding="UTF-8"?>
<fulldoc>
<address>
<street>building 6</street>
<city>BALTIMORE</city>
<state>MD</state>
<country>USA</country>
</address>
<filepath>
/sy/ase1501sunbox/EFTS-
15_0/collections
/made_using_mkvdv/xml_data/row2_well_formed.xml
</filepath>
</fulldoc>
')
go
(1 row affected)
quit
```


片段 5

此片段将重新创建文本索引。

```
isql -Sase1501sunbox -Usa -P -w2048 Dxml_db1
sp_create_text_index 'textsvr'
ti_address_tbl', 'address_tbl', " ", 'xmlcol'
go
(return status = 0)
quit
```

片段 6

此片段发出 **select** 请求，以便使用文本索引根据 XML 数据来分离行。

```
isql -Sase1501sunbox -Usa -P -w2048 -Dxml_db
select
    t1.id
from
    ti_address_tbl t1,
    address_tbl 2
where
    t1.id = t2.id and
    t1.index_any = 'USA <in> address'
go
quickpass disabled, reason: local table
SELECT
    id
FROM
    xml_db.dbo.ti_address_tbl
WHERE
    (
        index_any = 'USA <in> address'
    )
id
-----
1
2
(2 rows affected)
quit

isql -Sase1501sunbox -Usa -P -w2048 -Dxml_db
select
    t1.id
```

```
from
    ti_address_tbl t1,
    address_tbl t2
where
    t1.id=t2.id and
    t1.index_any = 'USA <in> country'
go
quickpass disabled, reason: local
table
SELECT
    id
FROM
    xml_db.dbo.ti_address_tbl
WHERE
    (
        t1.index_any = 'USA <in> country'
    )
id
-----
1
2
(2 rows affected)
quit
```

```
isql -Sase1501sunbox -Usa -P -w2048 -Dxml_db
select
    t1.id
from
    ti_address_tbl t1,
    address_tbl t2
where
    t1.id=t2.id and
    t1.index_any='USA'
go
quickpass disabled, reason: local table
SELECT
    id
FROM
    xml_db.dbo.ti_address_tbl
WHERE
    (
        t1.index_any = 'USA'
    )
id
-----
```

```
1
2
(2 rows affected))
quit
```

```
isql -Sase1501sunbox -Usa -P -w2048 -Dxml_db
select
    t1.id
from
    ti_address_tbl t1,
    address_tbl t2
where
    (
        t1.id=t2.id and
        t1.index_any='mailstop <in> address <and>
        MA <in>state'
    )
go
quickpass disabled, reason: local table
SELECT
    id
FROM
    xml_db.dbo_ti_address_tbl
WHERE
    (
        index_any ='mailstop <in> address <and> MA
        <in> state'
    )
id
-----
1
(1 row affected))
quit
```

```
isql -Sase1501sunbox -Usa -P -w2048 -Dxml_db
select
    t1.id
from
    to_address_tbl t1
    address_tbl t2
where
```

```
t1.id=t2.id and
t1.index_any =
'sy <in> filepath <and>
asesunbox <in> filepath <and>
EFTS-15_0 <in> filepath
<and> collections <in> filepath <and>
made_using_mkvdv <in> filepath <and>
xml_data <in> filepath <and> ro* <in> filepath'
go
quickpass disabled, reason: local table
SELECT
    id
FROM
    xml_db.dbo.ti_address_tbl
WHERE
    (
        index_any = 'sy <in> filepath <and>
ase1501sunbox <in> filepath <and>
EFTS-15_0 <in> filepath
<and> collections <in> filepath <and>
made_using_mkvdv <in> filepath <and>
xml_data <in> filepath <and> ro* <in> filepath'
id )
-----
1
2
(2 rows affected)
quit
```

```
isql -Sase1501sunbox -Usa -P -w2048 -Dxml_db
select
    t1.id
from
    ti_address_tbl t1,
    address_tbl 2
where
    t1.id = t2.id and
    t1.index_any =
        'sy <in> filepath <and>
ase1501sunbox <in> filepath <and>
```

```
EFTS-15_0 <in> filepath
<and> collections <in> filepath <and>
made_using_mkvdK <in> filepath <and>
xml_data <in> filepath <and> row2* <in> filepath'
go
quickpass disabled, reason: local table
SELECT
    id
FROM
    xml_db.dbo.ti_address_tbl
WHERE
    (
        index_any = 'sy <in> filepath <and>
ase1501sunbox <in> filepath <and>
EFTS-15_0 <in> filepath <and>
collections <in> filepath <and>
made_using_mkvdK <in>
filepath <and>
row2* <in> filepath')
id
-----
2
(1 row affected)
quit
```


索引

符号

- ... (省略号) 在 SQL 语句中 xvi
- { } (大括号)
 - SQL 语句中 xv
- , (逗号)
 - SQL 语句中 xv
- [] (方括号)
 - SQL 语句中 xv
- () (小括号)
 - SQL 语句中 xv
- <> (尖括号), 将 Verity 运算符括在 48

英文

- accrue** 运算符 47, 49
- Adaptive Server
 - 处理全文本查询 8
 - 连接到增强型全文本搜索引擎 1
- and** 运算符 47, 50
 - 与 **not** 修饰符 57
- auto_online** 配置参数 20, 64, 65, 146
- backDir** 配置参数 64, 65, 71, 141
- batch_blocksize** 配置参数 62
- batch_size** 配置参数 62, 65
 - 和性能 76
- charset** 配置参数 63, 65
 - 设置缺省值 66
- cis cursor rows** 配置参数 75
- cis packet size** 配置参数 75
- cluster_effort** 配置参数 46, 64, 65
 - 值 137
- cluster_keywords** 伪列 42, 46
- cluster_max** 配置参数 46, 64, 65
 - 值 137
- cluster_number** 伪列 42, 46

- cluster_order** 配置参数 46, 64, 65
 - 值 137
- cluster_style** 配置参数 46, 63, 65
 - 值 137
- collDir** 配置参数 63, 65
- complement** 运算符 47, 50
- default_Db** 配置参数 63, 65
- delete** 操作
 - 创建触发器 20
- dump database** 命令
 - 和 **sp_text_dump_database** 系统过程 71, 141
- Enhanced Full-Text Search Specialty Data Store
 - 组件 3-7
- errorLog** 配置参数 63, 65
- forceplan**
 - 和强制连接次序 75
- getsend** 程序 152
- highlight** 伪列 42
- id** 伪列 5, 42
 - 和查询优化 75
 - 映射到源表中的 IDENTITY 列 17
- IDENTITY 列
 - 添加的示例 22
 - 添加唯一索引 17
 - 向现有源表中添加 17
 - 向源表添加 17
 - 以文本索引显示 132
 - 与索引表连接 5, 8
 - 在源表中 3
- in** 运算符 47, 50
- index_any** 伪列 42
 - 和查询优化 74
- insert** 操作
 - 创建触发器 20
- installevent** 安装脚本
 - 编辑 15
 - 使用 14
 - 使用的示例 22

- installtextserver** 安装脚本
 - 编辑 13
 - 和创建多个增强型全文本搜索引擎 77
 - 位置 13
- instsvr.exe** 实用程序 61
- interfaces** 配置参数 63, 65
- interfaces** 文件
 - 设置位置 63, 65
 - 在 runserver 文件中指定 59
- /keys** 修饰符 34
- knowledge_base** 配置参数 38, 64, 65
- language**
 - 设置缺省值 66
- language** 配置参数 63, 65
 - 设置缺省值 66
- like** 运算符 47, 51
 - 启用 QBE 规范中的文本字符 25
- list**
 - 关键字 34
- max_docs** 伪列 43
 - 用集群结果集 46
 - 和排序顺序 67
 - 和提高查询性能 74
- max_indexes** 配置参数 63, 65
- max_packet_size** 配置参数 63, 65
- max_session_fd** 63, 65
- max_sessions** 配置参数 63, 65
 - 和性能 76
- max_stacksize** 配置参数 63, 65
- max_threads** 配置参数 63, 65
- min_sessions** 配置参数 63, 65
 - 和性能 76
- mksyid** 实用程序
 - 并创建自定义同义词库 34
 - 并检查缺省同义词库 33
- mktopics** 实用程序 37
- near** 运算符 47, 51, 52
- near/n** 运算符 47, 52
 - 与 **order** 修饰符 57
- nocase** 配置参数 64, 65, 69
- Open Server 跟踪标志 69
- Open Server 事件, 跟踪 69
- or** 运算符 47, 50
 - 与 **not** 修饰符 57
- paragraph** 运算符 47, 52
 - 与 **many** 修饰符 57
 - 与 **order** 修饰符 57
- phrase** 运算符 47, 52
 - 与 **many** 修饰符 57
- product** 运算符 47, 53
- QBE 规范
 - 请参见 按示例查询
- RPC 事件, 记录 68
- RPC。
 - 请参见 远程过程调用
- runserver 文件 59
 - 标志 59
- score** 伪列 5, 43–44
 - 和 **many** 修饰符 57
 - 用集群结果集 46
 - 排序依据 44
 - 和缺省排序顺序 67
- score** 值
 - Sybase 如何报告 43
- sentence** 运算符 47, 53
 - 与 **many** 修饰符 57
 - 与 **order** 修饰符 57
- showplan**
 - 和检查连接次序 74
- sort_by** 伪列 43
 - 和请求集群结果集 46
 - 和指定排序顺序 44
 - 并设置一个已定义的列作为排序规范 28
- sort_order** 配置参数 44, 63, 65, 67
- sp_addserver** 系统过程 78
- sp_check_text_index** 系统过程 126
- sp_clean_text_events** 系统过程 127
- sp_clean_text_indexes** 系统过程 128
- sp_create_text_index** 系统过程 18, 129–130
 - 创建使用过滤器的索引 30
 - 使用的示例 23
 - 指定多列 19
- sp_drop_text_index** 系统过程 131
- sp_help_text_index** 系统过程 132
- sp_optimize_text_index** 系统过程 73, 133
- sp_redo_text_events** 系统过程 134

- sp_refresh_text_index** 系统过程 135
 - 修改集合中的数据 20
 - 自动运行 20
- sp_show_text_online** 系统过程 136
- sp_statistics** 系统过程
 - 禁用 68, 73
- sp_text_cluster** 系统过程 137–138
- sp_text_configure** 系统过程 64, 139
- sp_text_dump_database** 系统过程 71, 140–142
- sp_text_kill** 系统过程 143
- sp_text_load_index** 系统过程 72, 144
- sp_text_notify** 系统过程 145
 - 关闭优化 73
 - 和性能问题 77
 - 修改集合中的数据 20
- sp_text_online** 系统过程 20, 146
 - 示例 23
- sp_traceoff** 远程过程调用 68, 74
- sp_traceon** 远程过程调用 68, 74
- SQL 分析, 跟踪 68
- SQL 语句中的符号 xv
- SQL 语句中的省略号 (...) xvi
- srv_traceflags** 配置参数 63, 65, 69
- startserver** 实用程序 59
- stem** 运算符 47, 53
 - 与 **many** 修饰符 57
- style.dft* 文件 30
- style.prm* 文件
 - 编辑现有集合 130
 - 编辑主文件 26
 - 并启用 Verity 功能 25
 - 现有集合的位置 27
 - 针对现有集合进行编辑 27
 - 主文件位置 26
- style.ufl* 文件 28, 30
- style.vgw* 文件 28, 30
- sum** 运算符 47, 54
- summary* 伪列 43
 - 使用 45
 - 使用前启用 25
- Sybase Central, 启动 60
- synonyms**
 - 语句 33
- syssservers* 表
 - 添加增强型全文搜索引擎 78
- TDS 标识, 跟踪 69
- TDS 标头, 跟踪 69
- TDS 数据, 跟踪 69
- text_db* 数据库 4, 70, 71
 - 和 *vesaux* 表 4
 - 和 *vesauxcol* 表 5
 - 备份 140
 - 从备份恢复 72
 - 更改名称 13, 15
- text_events* 表 6, 70
 - 和 **sp_text_dump_database** 71, 141
 - 和 **sp_text_load_index** 72
 - 创建 14
 - 创建示例 22
 - 从备份恢复 70, 72
 - 更改条目状态 134
 - 记录插入、更新和删除 135
 - 列 6
 - 删除条目 127
- textsrv.cfg* 文件
 - 示例 147
- thesaurus** 运算符 47, 54
 - 使用自定义同义词库 32
- topic** 运算符 39, 47, 55
- traceflags** 配置参数 63, 65
- Unicode 153
 - 和通配符搜索 68
- Unicode 支持 153
- update statistics**
 - 禁用 74
- update** 操作
 - 创建触发器 20
- user
 - 会话 76
 - 连接 76
- vdkCharset** 配置参数 63, 65
 - 设置缺省值 66
- vdkHome** 配置参数 63, 65
- vdkLanguage** 配置参数 63, 65
 - 设置缺省值 66

Verity

- 跟踪 Verity 处理 68
- 设置 Verity 目录 63

Verity 查询。

请参见 全文本搜索查询

Verity 集合。

请参见 集合

Verity 中的命令。

请参见 运算符 (命令)

vesaux 表

- 创建条目 130
- 更新 18
- 列 4
- 删除条目 128
- 在删除文本索引时删除条目 131

vesauxcol 表

- 更新 18
- 列 5
- 在删除文本索引时删除条目 131

wildcard 运算符 47, 55

- 使用 Unicode 格式数据 68
- 与 **case** 修饰符 57
- 与 **many** 修饰符 57

Windows NT

- 目录路径 xiv

word 运算符 47, 56

- 与 **case** 修饰符 57
- 与 **many** 修饰符 57

writetext 命令, 使用触发器 21

XML 数据, 针对字段和区域设置格式 155

XML 数据, 针对字段和区域设置格式, 示例 156

yesno 运算符 47, 56

A

按示例查询

- 和 **like** 运算符 51
- 启用 25
- 针对单个表配置 27
- 针对所有表配置 26

B

- 保持完整性 4
- 备份操作, 跟踪 68
- 备份和恢复 70
- 备份文件
 - 缺省位置 64, 65

C

参数

- 搜索 5

插入

- 和更新文本索引 6

查询

- 和伪列 5

查询, 全文本搜索

- 处理 8
- 排序顺序规范 44
- 请求集群结果集 46
- 和区分大小写 49
- 确保连接次序正确 74
- 使数据库联机 20
- 使用替代语法 49
- 提高性能 74–75
- 并使用主题 39
- 组件 41
- 处理全文本搜索 8
- 错误记录 68
- 错误日志文件
 - 设置路径名 63
 - 在 runserver 文件中指定 59

D

大括号 ({})

- SQL 语句中 xv

定义多个增强型全文本搜索引擎 13

逗号 (,)

- SQL 语句中 xv

多用户 79

F

复制文本索引 21

G

跟踪标志 68
 Open Server 69
 启用跟踪标志 11 和 12 73
 设置以检查连接次序 74
 更新
 和更新文本索引 6
 更新索引 73
 关闭增强型全文本搜索引擎 62
 关注事件, 跟踪 68
 Open Server 69
 过程。
 请参见 系统过程
 过滤器, 文档 4
 创建 30
 和文档区域 51

H

恢复 70
 将一个文本索引与源表同步 134
 会话, 用户数 76

J

集合 4, 71
 另请参见 文本索引
 备份 140
 创建 129
 从备份恢复 70, 72
 更新时的性能问题 77
 禁用优化 68, 73
 缺省语言 66
 缺省字符集 66
 删除 131
 设置位置 63
 位置 4

显示名称 132
 修改数据 20
 用数据填充 18
 优化 133
 和重建索引 134

集群 45
 编写查询 46
 在排序规范中 44
 配置参数的值 137
 启用 25
 设置 46
 修改参数值 137
 针对单个表配置 27
 针对所有表配置 26
 尖括号 (<>), 将 Verity 运算符括在 48
 将更改传播到集合 6
 脚本, 示例
 sample_text_main.sql 16, 21, 151
 接口
 跟踪增强型全文本搜索引擎与 Verity 之间的调用
 68

K

括号。
 请参见 方括号 [] 和尖括号 <>

L

联机数据库。
 请参见 数据库, 使联机
 连接, 用户数 76
 连接次序
 确保正确 74
 连接到增强型全文本搜索引擎 78
 连接源表和文本索引 3, 5, 8, 17, 41
 提高性能 74
 列
 有效的索引数据类型 3

M

命名增强型全文本搜索引擎 63, 65

P

排列文档。

请参见 相关性排列

排序规范

设置要作为排序依据的已定义列 28

排序顺序

max_docs 和排序顺序 67

按 *score* 44

和集群结果集 44, 46

按列 28, 44

设置缺省值 67

按时间戳 44, 67

在查询中 44

配置参数 62–64, 65

另请参见 各个配置参数

auto_online 146

backDir 71, 141

batch_size 参数和性能 76

charset 66

cluster_effort 46, 137

cluster_max 46, 137

cluster_order 46, 137

cluster_style 46, 137

language 66

max_sessions 参数和性能 76

min_sessions 参数和性能 76

nocase 69

sort_order 44, 67

srv_traceflags 69

vdkCharset 66

vdkLanguage 66

显示值 139

修改值 139

配置参数, Adaptive Server

cis cursor rows 75

cis packet size 75

配置文件

示例 147

Q

启动

和设置用户连接数 76

启动命令

和 *runserver* 文件 59

在 Windows NT 上 61

启动增强型全文本搜索引擎

UNIX 平台上 59

从 Sybase Central 60

作为服务 61

在 Windows NT 上 60–62

区分大小写

在 SQL 中 xv

在查询中 49

设置增强型全文本搜索引擎 69

区域。

请参见 文档区域

全文本搜索查询

处理 8

排序顺序规范 44

和请求集群结果集 46

和区分大小写 49

使数据库联机 20

使用替代语法 49

并使用主题 39

组件 41

S

删除

从 *text_events* 表 127

从 *vesaux* 表 128

和更新文本索引 6

时间戳

排序依据 67

使用跟踪标志记录事件 68–69

示例程序 **getsend** 152

示例脚本

sample_text_main.sql 16, 21, 151

- 示例文件
 - 配置文件 147
 - 说明按示例查询 152
 - 说明汇总 152
 - 说明集群 152
 - 说明主题功能 36, 152
 - 说明自定义同义词库 32, 152
- 事件, 记录 68–69
- 数据库
 - 使全文本搜索联机进行 20
- 数据库, 联机
 - 自动 64, 65
- 数据类型
 - 和索引 18
 - 索引列的 3, 129
- 搜索参数 5
- 索引表
 - 和 *id* 列 17
 - 创建 18, 129
 - 内容 5
 - 删除 131
 - 与源表连接 5
 - 在查询中 8

T

- 同义词库, 自定义 32
 - 和 **mk_syd** 实用程序 34
 - 并创建控制文件 33
 - 并检查缺省同义词库 33
 - 并替换缺省同义词库 35

W

- 完整性, 保持 4
- 网络请求, 跟踪 69
- 唯一索引
 - 创建示例 23
 - 添加到 **IDENTITY** 列 17

- 伪列 5
 - cluster_keywords* 42, 46
 - cluster_number* 42, 46
 - highlight* 42
 - id* 42
 - index_any* 42
 - max_docs* 43, 46
 - score* 43–44
 - sort_by* 43, 44, 46
 - summary* 43, 45
 - 在查询中 8
- 文本索引 71, 72
 - 备份 140
 - 创建 18, 129
 - 创建和批大小 76
 - 创建示例 22–23
 - 包含多列 19
 - 放到多个增强型全文本搜索引擎上 77
 - 复制 21
 - 和跟踪信息 68
 - 更新 73
 - 更新时的性能问题 77
 - 联机显示 136
 - 删除 131
 - 设置备份文件的位置 64, 65
 - 使联机 146
 - 使用文档过滤器 30
 - 和索引表 5
 - 显示列表 132
 - 用 *text_events* 表更新 6
 - 优化 133
 - 元数据 4
 - 在增强版中从备份恢复 70
 - 和重建索引 134
- 文本文档, 类型 4
- 文档过滤器 4
- 文档区域
 - 和文本索引中的多列 19
 - 使用 **in** 运算符 50
- 文件描述符
 - 设置限制 63

X

系统表

更新 125

系统过程

另请参见各系统过程

sp_check_text_index 126
sp_clean_text_events 127
sp_clean_text_indexes 128
sp_create_text_index 129–130
sp_drop_text_index 131
sp_help_text_index 132
sp_optimize_text_index 133
sp_redo_text_events 134
sp_refresh_text_index 135
sp_show_text_online 136
sp_text_cluster 137–138
sp_text_configure 139
sp_text_dump_database 140–142
sp_text_kill 143
sp_text_load_index 144
sp_text_notify 145
sp_text_online 146

限制

文件描述符 63

相关性排列 43–44

另请参见 score 伪列

小括号 ()

SQL 语句中 xiv

性能和调优

和 **sp_text_notify** 77

和使用多个增强型全文本搜索引擎 77

禁用文本索引优化 73

提高查询性能 74–75

添加唯一索引 17

重新配置 Adaptive Server 75

重新配置增强型全文本搜索引擎 76

Y

已处理的事件

从 *text_events* 表中删除 127

用户表。

请参见 源表

用户数据库 70, 71

备份 140

从备份恢复 72

联机显示 136

使全文本搜索联机进行 20, 146

显示文本索引列表 132

自动联机 64, 65

优化, 禁用 68, 73

游标事件, 记录 68

语法, 替代的 Verity 49

语约定, Transact-SQL xiv

语言事件, 记录 68

元数据 4

源表

将 IDENTITY 列添加到 17

内容 3

数据更改 135, 145

并显示文本索引 132

在查询中 8

远程过程调用

sp_traceoff 68, 74**sp_traceon** 68, 74

约定

另请参见 语法

目录路径 xiv

在手册中使用 xiv

运算符 (命令) 47–56

accrue 47, 49**and** 47, 50**complement** 47, 50**in** 47, 50**like** 47, 51**near** 47, 51, 52**near/n** 47, 52**or** 47, 50**paragraph** 47, 52**phrase** 47, 52**product** 47, 53**sentence** 47, 53**stem** 47, 53**sum** 47, 54**thesaurus** 47, 54**topic** 47, 55**wildcard** 47, 55**word** 47, 56

- yesno** 47, 56
 - 括在尖括号中 48
 - 和相关性排列 43–44
- 运算符的 **case** 修饰符 57
- 运算符的 **many** 修饰符 57
- 运算符的 **not** 修饰符 57
- 运算符的 **order** 修饰符 57
- 运算符修饰符
 - case** 57
 - many** 57
 - not** 57
 - order** 57
- 运行 **sp_refresh_text_index** 的触发器 20

Z

- 增强型全文本搜索引擎
 - 更改名称 13
 - 关闭 62
 - 连接到 78
 - 配置多个引擎 13, 77–78
 - 如何处理查询 8
 - 通知 *text_events* 表的更新 145
 - 文档过滤器 4
 - 用 Sybase Central 启动 60
 - 运算符 47–56
 - 在 UNIX 平台上启动 59
 - 在 Windows NT 上启动 60–62
 - 组件之间的关系 7
 - 作为服务启动 61
- 摘要
 - 编写查询请求 45
 - 启用 25
 - 针对单个表配置 27
 - 针对所有表配置 26
- 针对字段和区域设置 XML 数据格式的示例 156
- 知识库映射
 - 创建 38
 - 定义位置 38
- 智能分类器 37
- 中括号 []
 - SQL 语句中 xv
- 主题
 - 创建大纲文件 36
 - 创建复杂关系 37
 - 创建知识库映射 38
 - 创建主题组目录 37
 - 故障排除 39
 - 示例文件 36
 - 说明 35
 - 执行查询 39
- 主题的大纲文件 36
- 主题组目录 37
 - 映射到 38
- 自定义同义词库 32
 - 和 **mksyd** 实用程序 34
 - 并创建控制文件 33
 - 并检查缺省同义词库 33
 - 并替换缺省同义词库 35
- 自定义同义词库的同义词列表 33
- 字段
 - 设置 XML 数据的格式, 示例 156
 - 字段, 设置 XML 数据的格式 155
- 字符集
 - 设置缺省值 66
- 组件集成服务
 - 连接到增强型全文本搜索引擎 1
- 作为源表的代理表 3
- 作为源表的远程表 3

