



# **UltraLite®**

## **数据库管理和参考**

**2009 年 2 月**

**11.0.1 版**

## 版权和商标

版权所有 © 2009 iAnywhere Solutions, Inc. 部分版权所有 © 2009 Sybase, Inc. 保留所有权利。

本文档按原样提供，并不做任何形式的担保或承担任何责任（除非在您与 iAnywhere 达成的书面协议中另行规定）。

对本文档（全部或部分）的使用、打印、复制和分发须符合下列条件：1) 必须在整个或部分文档的所有副本中保留此声明和所有其它所有权声明，2) 不得修改本文档，3) 不得以任何形式表明您或 iAnywhere 之外的任何人是本文档的作者或提供者。

iAnywhere®、Sybase® 以及在 <http://www.sybase.com/detail?id=1011207> 上所列出商标均为 Sybase, Inc. 或其子公司的商标。® 表示在美国注册。

文中提及的所有其它公司和产品名可能是与其相关的各个公司的商标。

---

---

# 目录

关于本手册 .....	<b>xi</b>
关于 SQL Anywhere 文档 .....	xii
<b>UltraLite 简介 .....</b>	<b>1</b>
UltraLite 功能比较 .....	2
UltraLite 限制 .....	6
UltraLite 数据体系结构 .....	8
UltraLite 存储和文件名约定 .....	9
UltraLite 事务和状态管理 .....	11
UltraLite 隔离级别 .....	15
<b>实现 UltraLite 解决方案 .....</b>	<b>17</b>
UltraLite 支持的平台和协议 .....	18
选择 UltraLite 数据管理组件 .....	19
选择 UltraLite 编程接口 .....	20
<b>使用 UltraLite 数据库 .....</b>	<b>23</b>
创建和配置 UltraLite 数据库 .....	25
创建 UltraLite 数据库 .....	26
为 UltraLite 选择数据库创建参数 .....	30
UltraLite 字符集 .....	32
保护 UltraLite 数据库 .....	35
连接到 UltraLite 数据库 .....	37
UltraLite 数据库连接参数 .....	38
UltraLite 用户验证 .....	44
使用 ULSQLCONNECT 环境变量存储 UltraLite 参数 .....	46
将 UltraLite 部署到设备 .....	47
部署带 UltraLite 引擎的多个 UltraLite 应用程序 .....	49
通过 AES_FIPS 数据库加密部署 UltraLite .....	50
通过启用 TLS 的同步部署 UltraLite .....	51

部署 UltraLite HotSync 管道 .....	53
为 UltraLite 部署 ActiveSync 提供程序 .....	55
使用 ActiveSync 管理器注册应用程序 .....	57
部署 UltraLite 模式升级 .....	58
使用 UltraLite 数据库 .....	61
使用 UltraLite 表和列 .....	62
使用 UltraLite 索引 .....	71
使用 UltraLite 发布 .....	75
使用 UltraLite 用户 .....	79
使用事件通知 .....	81
SQL 直通的 UltraLite 支持 .....	83
UltraLite CustDB 示例 .....	85
CustDB 示例文件位置 .....	87
第 1 课: 构建和运行 CustDB 应用程序 .....	89
第 2 课: 登录和填充 UltraLite 远程数据库 .....	91
第 3 课: 使用 CustDB 客户端应用程序 .....	92
第 4 课: 与 CustDB 统一数据库同步 .....	94
第 5 课: 浏览 MobiLink 同步脚本 .....	96
构建您自己的应用程序 .....	97
UltraLite 性能与优化 .....	99
使用索引扫描 .....	100
确定优化程序使用的访问方法 .....	102
使用索引散列调整查询性能 .....	103
选择最佳的散列大小 .....	105
设置最大散列大小 .....	107
管理临时表 .....	108
刷新单个或分组的事务 .....	110
数据库的加密和模糊处理怎样影响性能 .....	111
UltraLite 优化策略 .....	112
<b>UltraLite 用作 MobiLink 客户端 .....</b>	<b>113</b>
UltraLite 客户端 .....	115
UltraLite 的内置同步功能 .....	116
自定义 UltraLite 客户端同步行为 .....	117
UltraLite 中的主键唯一性 .....	119

设计 UltraLite 中的同步 .....	123
使用 MobiLink 文件传输 .....	129
使用 UltraLite 的 ActiveSync 和 HotSync .....	131
Palm OS 上的 HotSync .....	132
Windows Mobile 上的 ActiveSync .....	135
UltraLite 同步参数和网络协议选项 .....	137
UltraLite 的同步参数 .....	138
UltraLite 同步流的网络协议选项 .....	161
<b>UltraLite 数据库参考 .....</b>	<b>163</b>
UltraLite 创建参数 .....	165
UltraLite 大小写创建参数 .....	167
UltraLite checksum_level 创建参数 .....	168
UltraLite 归类创建参数 .....	170
UltraLite date_format 创建参数 .....	171
UltraLite date_order 创建参数 .....	174
UltraLite fips 创建参数 .....	176
UltraLite max_hash_size 创建参数 .....	177
UltraLite nearest_century 创建参数 .....	179
UltraLite obfuscate 创建参数 .....	181
UltraLite page_size 创建参数 .....	182
UltraLite 精度创建参数 .....	184
UltraLite 小数位数创建参数 .....	185
UltraLite time_format 创建参数 .....	187
UltraLite timestamp_format 创建参数 .....	189
UltraLite timestamp_increment 创建参数 .....	192
UltraLite utf8_encoding 创建参数 .....	193
UltraLite 数据库属性 .....	195
访问 UltraLite 数据库属性 .....	199
UltraLite 数据库选项 .....	201
UltraLite commit_flush_count 选项 [临时性] .....	202
UltraLite commit_flush_timeout 选项 [临时性] .....	203
UltraLite global_database_id 选项 .....	204
UltraLite ml_remote_id 选项 .....	205
变更 UltraLite 持久性数据库选项设置 .....	206

UltraLite 连接参数 .....	207
UltraLite CACHE_SIZE 连接参数 .....	208
UltraLite CE_FILE 连接参数 .....	209
UltraLite COMMIT_FLUSH 连接参数 .....	210
UltraLite CON 连接参数 .....	211
UltraLite DBF 连接参数 .....	212
UltraLite DBKEY 连接参数 .....	214
UltraLite DBN 连接参数 .....	215
UltraLite MIRROR_FILE 连接参数 .....	216
UltraLite NT_FILE 连接参数 .....	218
UltraLite ORDERED_TABLE_SCAN 连接参数 [不建议使用] .....	219
UltraLite PALM_ALLOW_BACKUP 连接参数 .....	220
UltraLite PALM_FILE 连接参数 .....	221
UltraLite PWD 连接参数 .....	223
UltraLite RESERVE_SIZE 连接参数 .....	224
UltraLite START 连接参数 .....	225
UltraLite UID 连接参数 .....	226
UltraLite 实用程序 .....	227
支持的退出代码 .....	228
UltraLite 的 Interactive SQL 实用程序 (dbisql) .....	229
UltraLite 实用程序的 SQL 预处理器 (sqlpp) .....	232
UltraLite 创建数据库实用程序 (ulcreate) .....	235
Palm OS 的 UltraLite 数据管理实用程序 (ULDBUtil) .....	238
UltraLite 引擎实用程序 (uleng11) .....	240
UltraLite 引擎停止实用程序 (ulstop) .....	241
UltraLite 消除数据库 (ulerase) .....	242
Palm OS 的 UltraLite HotSync 管道安装实用程序 (ulcond11) .....	243
UltraLite 信息实用程序 (ulinfo) .....	245
UltraLite 初始化数据库实用程序 (ulinit) .....	249
UltraLite 装载 XML 到数据库实用程序 (ulload) .....	252
UltraLite 同步实用程序 (ulsync) .....	255
同步配置文件选项 .....	258
UltraLite 卸载旧数据库实用程序 (ulunload) .....	260
UltraLite 卸载旧数据库实用程序 (ulunloadold) .....	263
UltraLite 校验数据库实用程序 (ulvalid) .....	265

UltraLite 系统表 .....	267
查看或隐藏 UltraLite 系统表 .....	268
systable 系统表 .....	269
syscolumn 系统表 .....	270
sysindex 系统表 .....	271
sysixcol 系统表 .....	273
syspublication 系统表 .....	274
sysarticle 系统表 .....	275
sysuldata 系统表 .....	276
<b>UltraLite SQL 参考 .....</b>	<b>277</b>
UltraLite SQL 元素 .....	279
UltraLite 中的关键字 .....	280
UltraLite 中的标识符 .....	281
UltraLite 中的字符串 .....	282
UltraLite 中的注释 .....	283
UltraLite 中的数字 .....	284
UltraLite 中的 NULL 值 .....	285
UltraLite 中的特殊值 .....	286
UltraLite 中的日期和时间 .....	289
UltraLite 中的数据类型 .....	290
UltraLite 中的表达式 .....	303
UltraLite 中的运算符 .....	316
UltraLite 中的变量 .....	319
UltraLite 中的执行计划 .....	320
UltraLite SQL 函数 .....	323
函数类型 .....	324
SQL 函数 (A-D) .....	330
SQL 函数 (E-O) .....	359
SQL 函数 (P-Z) .....	386
UltraLite SQL 语句 .....	421
UltraLite 语句类别 .....	423
UltraLite ALTER DATABASE SCHEMA FROM FILE 语句 .....	424
UltraLite ALTER PUBLICATION 语句 .....	425
UltraLite ALTER SYNCHRONIZATION PROFILE 语句 .....	426

UltraLite ALTER TABLE 语句 .....	428
UltraLite CHECKPOINT 语句 .....	431
UltraLite COMMIT 语句 .....	432
UltraLite CREATE INDEX 语句 .....	433
UltraLite CREATE PUBLICATION 语句 .....	435
UltraLite CREATE SYNCHRONIZATION PROFILE 语句 .....	436
UltraLite CREATE TABLE 语句 .....	440
UltraLite DELETE 语句 .....	444
UltraLite DROP INDEX 语句 .....	445
UltraLite DROP PUBLICATION 语句 .....	446
UltraLite DROP SYNCHRONIZATION PROFILE 语句 .....	447
UltraLite DROP TABLE 语句 .....	448
UltraLite FROM 子句 .....	449
UltraLite INSERT 语句 .....	451
UltraLite LOAD TABLE 语句 .....	452
UltraLite ROLLBACK 语句 .....	456
UltraLite SELECT 语句 .....	457
UltraLite SET OPTION 语句 .....	459
UltraLite START SYNCHRONIZATION DELETE 语句 .....	460
UltraLite STOP SYNCHRONIZATION DELETE 语句 .....	461
UltraLite SYNCHRONIZE 语句 .....	462
UltraLite TRUNCATE TABLE 语句 .....	464
UltraLite UNION 语句 .....	466
UltraLite UPDATE 语句 .....	467
<b>UltraLite 故障排除 .....</b>	<b>469</b>
<b>无法启动 UltraLite 引擎 .....</b>	<b>470</b>
<b>升级后无法连接数据库 .....</b>	<b>471</b>
<b>数据库损坏 .....</b>	<b>472</b>
<b>数据库大小不稳定 .....</b>	<b>473</b>
<b>将 ASCII 数据导入新数据库 .....</b>	<b>474</b>
<b>实用程序始终以前一版本运行 .....</b>	<b>475</b>
<b>结果集无法预测地更改 .....</b>	<b>476</b>
<b>UltraLite 引擎客户端错误，错误为 -764 .....</b>	<b>477</b>



<b>术语表 .....</b>	<b>479</b>
术语表 .....	481
<b>索引 .....</b>	<b>509</b>

---

---

# 关于本手册

## 主题

本手册介绍用于小型设备的 UltraLite 数据库系统。

## 读者

本手册适用于希望在嵌入式设备或移动设备的数据存储和同步过程中利用 UltraLite 关系数据库的高性能、资源效率、稳健性和安全性的所有开发人员。

## 关于 SQL Anywhere 文档

完整的 SQL Anywhere 文档以四种形式提供，但所包含信息均相同。

- **HTML 帮助** 联机帮助文档包含完整的 SQL Anywhere 文档，其中包括手册和 SQL Anywhere 工具的上下文相关帮助。

如果使用 Microsoft Windows 操作系统，则联机帮助文档以 HTML 帮助 (CHM) 格式提供。若要访问此文档，请选择 [开始] » [程序] » [SQL Anywhere 11] » [文档] » [联机手册]。

管理工具使用同一联机文档来实现帮助功能。

- **Eclipse** 在 Unix 平台上以 Eclipse 格式提供完整的联机帮助。要访问文档，请从 SQL Anywhere 11 安装的 *bin32* 或 *bin64* 目录下运行 *sadoc*。

- **DocCommentXchange** DocCommentXchange 是一个用于访问和讨论 SQL Anywhere 文档的社区。

使用 DocCommentXchange 可以执行以下任务：

- 查看文档
- 检查是否有用户对文档各部分所做出的阐明
- 提供建议和修正意见以在将来的版本中为所有用户改进文档

访问 <http://dcx.sybase.com>。

- **PDF** 整套 SQL Anywhere 手册会以一组 Portable Document Format (PDF) 文件的形式提供。您必须有 PDF 阅读器才能查看信息。要下载 Adobe Reader，请访问 <http://get.adobe.com/reader/>。

若要在 Microsoft Windows 操作系统上访问 PDF 文档，请选择 [开始] » [程序] » [SQL Anywhere 11] » 文档 » [联机手册 - PDF 格式]。

要在 Unix 操作系统上访问 PDF 文档，请使用 Web 浏览器打开 *install-dir/documentation/zh/pdf/index.html*。

## 关于文档集中的手册

SQL Anywhere 文档由以下手册组成：

- **SQL Anywhere 11 - 简介** 本手册介绍 SQL Anywhere 11，一个提供数据管理和数据交换技术的综合数据包，通过它可以为服务器环境、台式机环境、移动环境以及远程办公环境快速开发由数据库驱动的应用程序。
- **SQL Anywhere 11 - 更改和升级** 本手册介绍 SQL Anywhere 11 以及该软件以前版本中的新功能。
- **SQL Anywhere 服务器 - 数据库管理** 本手册介绍如何运行、管理及配置 SQL Anywhere 数据库。它介绍了数据库连接、数据库服务器、数据库文件、备份过程、安全性、高可用性、使用复制服务器进行复制以及管理实用程序和选项。

- **SQL Anywhere 服务器 - 编程** 本手册介绍如何使用 C、C++、Java、PHP、Perl、Python 和 .NET 编程语言（例如 Visual Basic 和 Visual C#）建立和部署数据库应用程序。其中介绍了各种编程接口，如 ADO.NET 和 ODBC。
- **SQL Anywhere 服务器 - SQL 参考** 本手册提供了系统过程和目录（系统表和视图）的参考信息。也介绍了 SQL 语言（搜索条件、语法、数据类型和函数）的 SQL Anywhere 实现。
- **SQL Anywhere 服务器 - SQL 的用法** 本手册介绍如何设计和创建数据库；如何导入、导出和修改数据；如何检索数据以及如何建立存储过程和触发器。
- **MobiLink - 入门** 本手册介绍基于会话的关系数据库同步系统 MobiLink。MobiLink 技术支持双向复制并且非常适用于移动计算环境。
- **MobiLink - 客户端管理** 本手册介绍如何设置、配置和同步 MobiLink 客户端。MobiLink 客户端可以是 SQL Anywhere 或者 UltraLite 数据库。本手册同时也介绍了 Dbmlsync API，通过它可以无缝地将同步集成到 C++ 或 .NET 客户端应用程序中。
- **MobiLink - 服务器管理** 本手册说明如何设置和管理 MobiLink 应用程序。
- **MobiLink - 服务器启动的同步** 本手册介绍 MobiLink 服务器启动的同步，这种功能允许 MobiLink 服务器启动同步或在远程设备上进行操作。
- **QAnywhere** 本手册介绍 QAnywhere，一个用于移动、无线、台式机和膝上型客户端的消息传递平台。
- **SQL Remote** 本手册介绍用于移动计算的 SQL Remote 数据复制系统，此系统支持使用电子邮件或文件传输等间接链接共享 SQL Anywhere 统一数据库和多个 SQL Anywhere 远程数据库之间的数据。
- **UltraLite - 数据库管理和参考** 本手册介绍适用于小型设备的 UltraLite 数据库系统。
- **UltraLite - C 及 C++ 编程** 本手册介绍 UltraLite C 和 C++ 编程接口。利用 UltraLite，可以开发数据库应用程序，并将它们部署到手持式设备、移动设备或嵌入式设备。
- **UltraLite - M-Business Anywhere 编程** 本手册介绍 UltraLite for M-Business Anywhere。利用 UltraLite for M-Business Anywhere，用户可以开发基于 Web 的数据库应用程序，并将它们部署到运行 Palm OS、Windows Mobile 或 Windows 的手持式设备、移动设备或嵌入式设备。
- **UltraLite - .NET 编程** 本手册介绍 UltraLite.NET。利用 UltraLite.NET，您可以开发数据库应用程序，并将它们部署到计算机、手持式设备、移动设备或嵌入式设备。
- **UltraLiteJ** 本手册介绍 UltraLiteJ。利用 UltraLiteJ，可以在支持 Java 的环境中开发和部署数据库应用程序。UltraLiteJ 支持 BlackBerry 智能手机和 Java SE 环境。UltraLiteJ 基于 iAnywhere UltraLite 数据库产品。
- **错误消息** 本手册提供了 SQL Anywhere 错误消息及其诊断信息的完整列表。

## 文档约定

本节列出了本文档中使用的约定。

## 操作系统

SQL Anywhere 可以在各种平台上运行。在大多数情况下，该软件在所有平台上的行为都是相同的，但也有变动或限制。这些变动或限制通常基于基础操作系统（Windows、Unix），很少基于特定变型（AIX、Windows Mobile）或版本。

为了简化对操作系统的提及，本文档按如下方式对支持的操作系统进行分组：

- **Windows** Microsoft Windows 系列包括 Windows Vista 和 Windows XP（主要用于服务器、台式计算机和膝上型计算机），以及 Windows Mobile（用于移动设备）。

除非另外指定，否则当本文档提及 Windows 时，是指所有基于 Windows 的平台，包括 Windows Mobile。

- **Unix** 除非另外指定，否则当本文档提及 Unix 时，是指所有基于 Unix 的平台，包括 Linux 和 Mac OS X。

## 目录和文件名

大部分情况下，对目录和文件名的引用在所有支持的平台上都是类似的，只需在不同形式之间进行简单的转换。这时需使用 Windows 约定。在细节更为复杂的情况下，文档显示所有相关形式。

下面是文档编写中用于简化目录和文件名的约定：

- **大写和小写目录名** 在 Windows 和 Unix 上，目录和文件名可以包括大写和小写字母。创建目录和文件时，文件系统会保留字母大小写。

在 Windows 上，对目录和文件的提及不区分大小写。混合使用大小写的目录和文件名很常见，但使用所有小写字母来提及目录和文件的形式也很常见。SQL Anywhere 安装包包含诸如 *Bin32* 和 *Documentation* 的目录。

在 Unix 上，对目录和文件的提及区分大小写。混合使用大小写的目录和文件名不常见。大多数的目录和文件名全部使用小写字母。SQL Anywhere 安装包包含诸如 *bin32* 和 *documentation* 的目录。

本文档采用 Windows 形式的目录名。大多数情况下，在 Unix 上可以将大小写混合形式的目录名转换成小写字母的等效目录名。

- **分隔目录和文件名的斜线** 文档使用反斜线作为目录分隔符。例如，PDF 格式的文档位于 *install-dir\Documentation\zh\PDF*（Windows 形式）。

在 Unix 上，用正斜线替换反斜线。PDF 文档位于 *install-dir/documentation/zh/pdf* 下。

- **可执行文件** 文档使用 Windows 约定显示可执行文件名（带有诸如 *.exe* 或 *.bat* 后缀）。在 Unix 上，可执行文件名没有后缀。

例如，在 Windows 上，网络数据库服务器是 *dbsrv11.exe*。在 Unix 上是 *dbsrv11*。

- **install-dir** 在安装过程中，选择 SQL Anywhere 的安装位置。创建环境变量 *SQLANY11*，用来表示此位置。文档中以 *install-dir* 表示此位置。

例如，本文档将此文件表示为 *install-dir\readme.txt*。在 Windows 上，这等同于 *%SQLANY11%\readme.txt*。在 Unix 上，这等同于 *SQLANY11/readme.txt* 或 *{SQLANY11}/readme.txt*。

有关 *install-dir* 缺省位置的详细信息，请参见“SQLANY11 环境变量”一节《SQL Anywhere 服务器 - 数据库管理》。

- **samples-dir** 在安装过程中，选择 SQL Anywhere 随附的示例的安装位置。创建环境变量 SQLANY11，用来表示此位置。文档中以 *samples-dir* 表示此位置。

要在 *samples-dir* 中打开 Windows 资源管理器窗口，请在 [开始] 菜单中，选择 [程序] » [SQL Anywhere 11] » [示例应用程序和项目]。

有关 *samples-dir* 缺省位置的详细信息，请参见“SQLANY11 环境变量”一节《SQL Anywhere 服务器 - 数据库管理》。

## 命令提示符和命令 shell 语法

大多数操作系统都提供一种或多种使用命令 shell 或命令提示符来输入命令和参数的方法。Windows 命令提示符包括 Command Prompt (DOS 提示符) 和 4NT。Unix 命令 shell 包括 Korn shell 和 bash。每个 shell 都具有一些功能，其能力不仅仅局限于简单命令。这些功能通过特殊字符来驱动。特殊字符和功能随 shell 的不同而不同。如果没有正确使用这些特殊字符，通常会导致语法错误或意外行为。

本文档以普通形式提供命令行示例。如果这些示例中包含 shell 的特殊字符，则命令需要根据特定 shell 进行修改。修改方法不在本文档所述范围之内，但通常是在包含这些特殊字符的参数两旁加上引号，或是在特殊字符前面使用转义字符。

下面是命令行语法的一些示例，不同的平台可能会有不同的形式：

- **括号和大括号** 有些命令行选项需要一个参数，该参数将以列表形式接受详细的值指定。该列表通常用括号或大括号括起来。本文档使用括号。例如：

```
-x tcpip(host=127.0.0.1)
```

如果括号导致出现语法问题，用大括号替代：

```
-x tcpip{host=127.0.0.1}
```

如果两种形式都将产生语法问题，应按照 shell 的要求，用引号将整个参数括起来：

```
-x "tcpip(host=127.0.0.1)"
```

- **引号** 如果必须在参数值中指定引号，该引号可能会与用于括参数的引号的传统用法发生冲突。例如，要指定值中包含双引号的加密密钥，则可能必须用引号括起密钥，然后转义嵌入的引号：

```
-ek "my \"secret\" key"
```

在许多 shell 中，密钥的值为 my "secret" key。

- **环境变量** 本文档介绍设置环境变量。在 Windows shell 中，环境变量使用语法 %ENVVAR% 来指定。在 Unix shell 中，环境变量使用语法 \$ENVVAR 或 \${ENVVAR} 来指定。

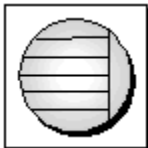
## 图标

本文档中使用了下列图标。

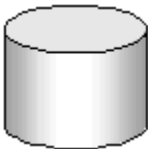
- 客户端应用程序。



- 数据库服务器，如 Sybase SQL Anywhere。



- 数据库。在某些高水平的图中，可以使用此图标表示数据库和管理该数据库的数据库服务器。



- 复制或同步中间件。用于帮助在数据库之间共享数据。例如 MobiLink 服务器和 SQL Remote 消息代理。



- 编程接口。



## 联系文档小组

我们欢迎您就本帮助文档提出意见、建议和反馈信息。

要提交意见和建议，请发送电子邮件到 SQL Anywhere 文档小组，地址为 [iasdoc@sybase.com](mailto:iasdoc@sybase.com)。虽然我们不对这些电子邮件进行回复，但您的反馈会帮助我们改进文档，因此我们真诚地欢迎您提出宝贵的意见和建议。



## DocCommentXchange

也可以使用 DocCommentXchange 将意见或建议直接置于帮助主题中。DocCommentXchange (DCX) 是一个用于访问和讨论 SQL Anywhere 文档的社区。使用 DocCommentXchange 可以执行以下任务：

- 查看文档
- 检查是否有用户对文档各部分所做出的阐明
- 提供建议和修正意见以在将来的版本中为所有用户改进文档

访问 <http://dcx.sybase.com>。

## 查找详细信息并请求技术支持

附加信息和资源可从 Sybase iAnywhere 开发人员社区获得，网址是 <http://www.sybase.com/developer/library/sql-anywhere-techcorner>。

如果您有问题或是需要帮助，可将邮件发布到下面所列的 Sybase iAnywhere 新闻组。

当您向这些新闻组发布邮件时，请务必提供问题的详细信息，包括 SQL Anywhere 版本的内部版本号。可以通过运行以下命令找到此信息：**dbeng11 -v**。 **dbeng11 -v**。

新闻组位于 *forums.sybase.com* 新闻服务器上。

这些新闻组包括：

- [sybase.public.sqlanywhere.general](#)
- [sybase.public.sqlanywhere.linux](#)
- [sybase.public.sqlanywhere.mobilink](#)
- [sybase.public.sqlanywhere.product\\_futures\\_discussion](#)
- [sybase.public.sqlanywhere.replication](#)
- [sybase.public.sqlanywhere.ultralite](#)
- [ianywhere.public.sqlanywhere.qanywhere](#)

有关 Web 开发问题，请访问 <http://groups.google.com/group/sql-anywhere-web-development>。

### 新闻组免责声明

iAnywhere Solutions 没有义务为其新闻组提供解决方案、信息或建议，除提供系统操作员监控服务和确保新闻组的运行和可用性外，iAnywhere Solutions 也没有义务提供任何其它服务。

如果时间允许，iAnywhere 技术顾问以及其他员工也会对新闻组服务提供帮助。他们是在自愿的基础上提供帮助的，所以可能无法定期提供解决方案和信息。他们可以提供多少帮助取决于他们的工作量。

---

---

# UltraLite 简介

## 目录

UltraLite 功能比较 .....	2
UltraLite 限制 .....	6
UltraLite 数据体系结构 .....	8
UltraLite 存储和文件名约定 .....	9
UltraLite 事务和状态管理 .....	11
UltraLite 隔离级别 .....	15

---

UltraLite 是一个精简的关系数据库，具有 SQL Anywhere 的许多特征和功能。UltraLite 可以作为用于企业范围的移动数据管理的 SQL Anywhere 解决方案的一部分进行安装，也可作为独立嵌入式解决方案的一部分进行安装。

UltraLite 提供内置功能来协调公司数据。当部署为 MobiLink 客户端时，可以实施同步解决方案，该解决方案以适时和可靠的方式传递任务关键信息。用户可以记录和访问他们需要的数据—即使当他们无法直接访问公司网络时。

应用程序开发接口允许您为用户编写应用程序和创建附加值。

## 另请参见

- [“了解 MobiLink 同步” 《MobiLink - 入门》](#)
- [“MobiLink 客户端” 《MobiLink - 客户端管理》](#)
- [MobiLink - 入门](#)
- [MobiLink - 客户端管理](#)
- [MobiLink - 服务器管理](#)

## UltraLite 功能比较

在 C/C++ 版本中，UltraLite 数据库和管理系统将应用程序大小增加 400-500 KB。SQL Anywhere 数据库、数据库服务器和同步客户端增加大约 6 MB。

功能	SQL Anywhere	UltraLite	注意事项
事务处理、参照完整性和多表连接	X	X	
触发器、存储过程和视图	X		
外部存储过程（可调用的外部 DLL）	X		
内置参照和实体完整性	X	X	声明的参照完整性（在此处级联删除和更新）是 UltraLite 数据库不支持的功能，除了同步过程中为此目的级联删除时。请参见“避免与外键循环相关的同步问题”一节第 126 页。
级联更新和删除	X	受限	
动态多数据库支持	X	X	仅对于 UltraLite 引擎。
多线程应用程序支持	X	X	
行级锁定	X	X	
XML 卸载和装载实用程序		X	UltraLite 使用单独的管理工具完成 XML 的装载和卸载。这不内置于运行时。请参见“UltraLite 装载 XML 到数据库实用程序 (ulload)”一节第 252 页和“UltraLite 卸载旧数据库实用程序 (ulunload)”一节第 260 页。
XML 导出和导入实用程序	X		SQL Anywhere 使用 SQL 语句将数据导出/导入到 XML。也可以使用 dbunload 导出数据。请参见“导入和导出数据”《SQL Anywhere 服务器 - SQL 的用法》。
SQLX 功能	X		

功能	SQL Anywhere	UltraLite	注意事项
SQL 函数	X	X	并非所有的 SQL 函数都可在 UltraLite 应用程序中使用。如果使用不受支持的函数，则会触发 [在 UltraLite 中功能不可用] 错误。请参见“UltraLite SQL 函数”第 323 页。
SQL 语句	X	受限	与 SQL Anywhere 相比，在 UltraLite 中 SQL 语句的范围受到限制。请参见“UltraLite SQL 语句”第 421 页。
集成 HTTP 服务器	X		
用于数据库文件和网络通信的高度加密	X	X	
事件调度和处理	X		
高性能、自行调优、基于开销的查询优化程序	X		UltraLite 拥有查询优化程序，但是它不像 SQL Anywhere 中的优化程序那样范围广泛。因此，UltraLite 优化程序在复杂查询中可能不能提供像 SQL Anywhere 优化程序那样的高性能。
几个线程安全的 API 的选择	X	X	UltraLite 为应用程序开发人员提供独特的灵活体系结构，允许创建用于变化的和/或不同的部署环境的应用程序。请参见“选择 UltraLite 编程接口”一节第 20 页。
游标支持	X	X	请参见“UltraLite 限制”一节第 6 页。
使用高级高速缓存管理系统动态调整高速缓存的大小	X		在 UltraLite 中，高速缓存大小的调整是静态的。尽管如此，在启动数据库时，UltraLite 允许设置高速缓存的大小，为您提供缩放高速缓存大小的功能。请参见“UltraLite CACHE_SIZE 连接参数”一节第 208 页。
系统或应用程序失败后数据库的恢复	X	X	
二进制大对象 (BLOB) 支持	X	X	UltraLite 不能索引或比较 BLOB。
Windows 性能监视器集成	X		

功能	SQL Anywhere	UltraLite	注意事项
联机表和索引碎片整理	X		
联机备份	X		
占用资源较少，可以小到 500 KB		X	小型设备的处理器速度往往相对较慢。UltraLite 使用专用于这些设备的算法和数据结构，因此 UltraLite 可继续提供很高的性能而且使用内存较少。
支持 Palm OS		X	
从桌面操作系统将设备直接连接到 Windows Mobile 设备。		X	允许桌面操作系统连接到 Windows Mobile 设备上部署的数据库之前，SQL Anywhere 数据库需要数据库服务器。在 UltraLite 上，只需要以 <b>WCE:\</b> 作为连接字符串的前缀。请参见“ <a href="#">Windows Mobile</a> ”一节第 41 页。
通过使用索引进行高性能更新和检索	X	X	UltraLite 使用一种机制确定是使用索引还是通过直接扫描行来搜索每张表。  此外，可以散列索引来提高数据检索速度。请参见“ <a href="#">UltraLite max_hash_size 创建参数</a> ”一节第 177 页。
同步到 Oracle、DB2、Sybase Adaptive Server Enterprise 或 SQL Anywhere	X	X	
内置同步		X	与 SQL Anywhere 部署不同，UltraLite 不需要客户端代理以进行同步。同步内置于 UltraLite 运行时以最小化需要部署的组件。请参见“ <a href="#">UltraLite 客户端</a> ”第 115 页。
进程中执行		X	
计算列	X		
声明的临时表/全局临时表	X		
系统函数	X		UltraLite 不支持 SQL Anywhere 系统函数，包括属性函数。无法将它们包含为 UltraLite 应用程序的一部分。

功能	SQL Anywhere	UltraLite	注意事项
时间戳列	X	X	SQL Anywhere Transact-SQL 时间戳列是使用缺省的 DEFAULT TIMESTAMP 创建的。  UltraLite 时间戳列是使用缺省的 DEFAULT CURRENT TIMESTAMP 创建的。因此，当更新行时，UltraLite 不自动更新时间戳。
确定基于对象的所有权及访问的基于用户的权限模式	X		UltraLite 主要设计用于不需要授权系统的单独用户数据库。但是，可以包含最多四个用户 ID 和口令，它们仅用于验证目的。这些用户可以访问所有数据库对象。请参见“ <a href="#">UltraLite 用户验证</a> ”一节第 44 页。

## UltraLite 限制

将 UltraLite 和 SQL Anywhere 的限制进行比较，请参见“SQL Anywhere 大小和数量限制”一节《SQL Anywhere 服务器 - 数据库管理》。

统计信息	UltraLite 的最大值
每个数据库的连接数	对单线程应用程序最多为 14 个。
打开的并发连接数	对于 Palm OS 最多为 8 个。 对于所有其它 OS 最多为 32 个。
每个应用程序的并发连接总数	在 Palm OS 上最多为 16 个。 对于所有其它 OS 最多为 64 个。
SQL 通信区	最多为 63。
基于文件的持久存储区（数据库大小）	2 GB 文件，或操作系统对文件大小的限制值。
Palm Computing Platform 数据库大小	128 MB（主存储区）。 2 GB（扩展卡文件系统）。
每张表中的行	最多为 1 千 6 百万个。 <sup>1</sup>
每个数据库中的行	受限于持久存储区。
表大小	仅受数据库大小的限制。
每个数据库中的表	仅受数据库大小的限制。
每个表中的列	行大小受限于页面大小，因此每张表中列数的实际限制是从该大小派生的。通常，实际限制远小于 4000。
每张表中的索引	仅受数据库大小的限制。
每个事务中引用的表	无限制。
存储过程长度	不适用。
每个数据库中的存储过程	不适用。



统计信息	UltraLite 的最大值
每个数据库中的触发器	不适用。
嵌套	不适用。
发布数	32 个发布。
页面大小	16 KB。
行大小	每个压缩行的长度必须不超过页面大小。请参见“ <a href="#">行压缩和表定义</a> ”一节第 62 页。 当字符串长度小于列大小时，在无填充情况下存储字符串。当单独存储这些字符串时，此限制会排除声明为 long binary 和 long varchar 的列。
每个连接的游标数	在一个给定的 UltraLite 数据库连接上所允许的最大游标数为 64（适用于所有平台）。
Long binary/long varchar 大小	仅受数据库大小的限制。

<sup>1</sup> 在某些情况下，行更改（删除和更新）和其它状态信息与行数据一起维护。此信息允许同步这些更改。因此，实际行限制可能比 1600 万小，这取决于同步之间表上的事务数或表是否已同步。请参见“[UltraLite 事务处理](#)”一节第 14 页。

## UltraLite 数据体系结构

UltraLite 是具有真正平台独立性的移动数据库。UltraLite 旨在创建用于小型设备（例如手机、手持式计算机和嵌入式设备）的自定义解决方案。

UltraLite 为您提供了一个完整的数据库管理系统，包括：

- **开发层** UltraLite 支持多个编程接口，避免将您锁定到一个部署平台、开发工具或 IT 基础结构产品集。

有关应选择哪个 API 的信息，请参见“[选择 UltraLite 编程接口](#)”一节第 20 页。

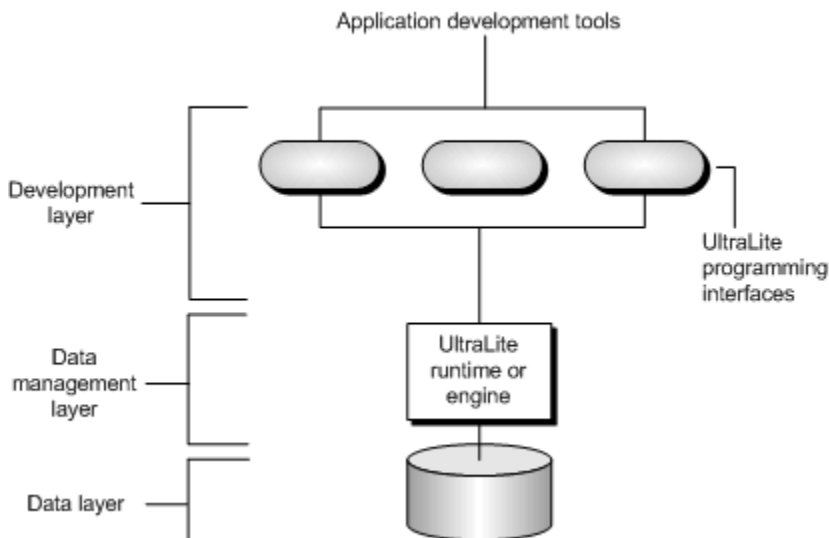
为了帮助您维护 UltraLite 项目，UltraLite 使用一组综合的管理工具完成它的开发支持。可以将这些工具作为命令行实用程序运行，也可以作为 Sybase Central 的 UltraLite 插件中的向导运行。

- **数据管理层和同步客户端** 可以使用称为运行库的进程中库或称为引擎的单独进程连接到 UltraLite 数据库。进程控制连接和数据访问请求。它们还包括一个链接 UltraLite 数据库和企业后端系统的内置双向同步客户端。

有关应选择哪个进程的信息，请参见“[选择 UltraLite 数据管理组件](#)”一节第 19 页。

- **数据层** 此层是作为文件或一系列数据记录（用于 Palm OS）存储的本地数据存储库。请参见“[UltraLite 存储和文件名约定](#)”一节第 9 页。

下图表示数据、管理和开发层。



## UltraLite 存储和文件名约定

设备的文件管理要求指定 UltraLite 数据库如何存储和必须使用何种数据库命名约定。大多数平台使用传统的基于文件的存储，而其它平台（例如 Palm OS 基于记录的存储）则要求以不同方式保存数据库。但是，UltraLite 数据库通常称为文件。根据平台限制，可以在开发桌面操作系统上创建数据库，然后将其部署到一个或多个平台。

### 另请参见

- “在 UltraLite 连接参数中指定文件路径” 一节第 40 页

## UltraLite 数据库模式

数据库的逻辑框架称作**模式**。在 UltraLite 中，模式是作为为 UltraLite 数据库保存元数据的系统表的目录来维护的。在系统表中存储的元数据包括：

- 索引定义。请参见“sysindex 系统表”一节第 271 页和“sysixcol 系统表”一节第 273 页。
- 表定义。请参见“systable 系统表”一节第 269 页。
- 列定义。请参见“syscolumn 系统表”一节第 270 页。
- 发布定义。请参见“syspublication 系统表”一节第 274 页和“sysarticle 系统表”一节第 275 页。
- 用户名和口令。请参见“sysuldata 系统表”一节第 276 页。

### 使用 DDL 语句更改模式

可以使用适当的数据定义语言（Data Definition Language，简称 DDL）语句更改数据库模式。如果您需要的不仅仅是几个更改，则可以在 SQL 脚本中使用 ALTER DATABASE SCHEMA FROM FILE 语句修改模式定义。

模式变更可能需要大量的时间。例如，关联表中的所有行必须在列类型变更时更新。当不存在以下任何情况时，DDL 语句会成功执行：

- 未提交的事务。
- 模式的其它活动使用（例如，同步、预准备但未发布的语句或执行数据库操作）。

如果两个条件中的一个为真，则 DDL 语句失败。执行 DDL 语句时，在 DDL 语句完成模式更改前，将阻塞使用数据库的任何其它尝试。

### 另请参见

- “UltraLite ALTER DATABASE SCHEMA FROM FILE 语句” 一节第 424 页
- “UltraLite SQL 语句” 第 421 页

## UltraLite 临时文件

在数据库操作过程中，除了数据库文件，UltraLite 还创建和维护 *临时文件*。不需要以任何方式使用或维护文件。

仅 UltraLite 使用临时文件并在与 UltraLite 数据库本身相同的文件路径（如果存在）中维护临时文件。临时文件具有与数据库相同的文件名，但是有以下差异：

- **对于基于文件的平台** 代字号包括在文件扩展名中。例如，如果运行 *CustDB.udb* 示例数据库，则在与该文件相同的目录中维护称为 *CustDB.~db* 的临时文件。
- **对于基于记录的平台** 在文件名的结尾附加代字号。例如，如果 *CustDB.udb* 作为 Palm OS 的基于记录的文件存在，则它的临时文件将作为 *CustDB.udb~* 维护。

### 提示

可以在没有数据丢失的情况下安全删除临时文件—只要 UltraLite 不在运行。它不包含会话间需要的信息。

## UltraLite 临时表

访问计划使用临时表在其执行过程中在瞬时性或临时工作表中存储数据。此表仅在访问计划执行时存在。通常，当需要在访问计划早期计算子查询时，或当中间结果无法放置在可用内存中时使用临时表。

临时表中的数据只针对一个连接而保存。临时表由行和列组成。每一列保存的是一类特定的信息（例如电话号码或名称），而每一行指定的是一个特定条目。

通过对用于 ORDER BY 或 GROUP BY 子句中的列使用索引可以避免使用临时表。

### 另请参见

- [“UltraLite 临时文件”一节第 10 页](#)
- [“UltraLite 性能与优化”第 99 页](#)
- [“查看执行计划时”一节第 320 页](#)

## UltraLite 事务和状态管理

UltraLite 同时维护状态信息和数据库中的数据。UltraLite 跟踪和存储状态信息以便可以管理：

- 并发连接，因此 UltraLite 可以根据需要共享资源。请参见“[UltraLite 中的并发](#)”一节第 11 页。
- 同步进度计数，以确保成功进行同步。请参见“[UltraLite 的内置同步功能](#)”一节第 116 页。
- 行状态，以通过跟踪数据如何在同步之间更改来维护数据完整性。请参见“[UltraLite 行状态](#)”一节第 12 页。
- 事务，以确定数据何时和如何提交。在 UltraLite 中，事务要么全部得到处理，要么根本不做处理。请参见“[UltraLite 事务处理](#)”一节第 14 页。
- 恢复和备份信息，以在操作系统崩溃、最终用户操作（例如删除存储卡）或在 UltraLite 运行的过程中重置设备时保护数据。请参见“[备份和恢复 UltraLite 中的数据](#)”一节第 13 页。

## UltraLite 中的并发

并发是 UltraLite 通过同时允许多个连接来共享资源的一种方式。UltraLite 使用以下方法管理并发性：

- **多个数据库** 单个 UltraLite 应用程序可以打开到多个数据库的连接。UltraLite 可以在 Palm OS 上打开多至 8 个数据库，对于其它所有平台，可以打开多至 32 个数据库。
- **多个应用程序** 一次仅可以由一个进程打开 UltraLite 数据库。如果打算在多个应用程序间支持并发，则选择 UltraLite 引擎作为数据管理组件。请参见“[选择 UltraLite 数据管理组件](#)”一节第 19 页。
- **多线程** UltraLite 支持多线程应用程序。可以编写一个使用多线程的应用程序，它的每个线程都可以连接到相同的或不同的数据库。

如果要使用运行库管理您的数据库，请确保没有超过 UltraLite 所支持的并发连接限制。这些限制为：

- Palm OS 支持 16 个并发连接。
- 其它所有平台支持 64 个并发连接。

如果使用 UltraLite 引擎管理数据库连接，则可使用的 SQLCA 数目通常限制为 128。但是，UltraLite.NET API 的实现将此限制明显减少到正在运行的 UltraLite.NET 客户端数目减去 128。

- **多个事务/请求** 每个连接在同一时刻可以有一个进行中的事务。事务可以由一个请求或多个请求组成。在提交事务之前，该事务期间所作的数据库修改在数据库中不会成为永久性的。某个事务中所作的数据库修改或者全部提交，或者全部回退。请参见“[UltraLite 事务处理](#)”一节第 14 页。
- **同步** 在上载和下载期间，允许对数据库进行读写访问。但是，如果应用程序更改了一行，而随后下载也尝试更改此行，则下载失败并回退。使用 `Disable Concurrency` 同步参数禁用同步期间的数据库访问。请参见“[Additional Parameters 同步参数](#)”一节第 138 页。

如果同步失败，UltraLite 在所有平台上支持断点续传下载。请参见“处理失败的下载”一节《MobiLink - 服务器管理》。

### 另请参见

- “UltraLite 客户端”第 115 页
- “Additional Parameters 同步参数”一节第 138 页

## UltraLite 行状态

维护行状态信息是 UltraLite 功能集的强大部分。对于数据同步来说，跟踪表和行的状态特别重要。

### 数据更改

内部标记用于跟踪 UltraLite 数据库中的行状态。行状态控制事务处理、恢复和同步。插入、更新或删除行时，该行的状态会被修改，以反映操作和执行操作的连接。提交事务时，会修改该事务影响到的所有行的状态，以反映该提交操作。如果在提交时发生了意外故障，则整个事务将会回退。以下列表总结了这些行为：

- **发出删除操作时** 每个受影响的行的状态均会更改，以反映该行已被删除。恢复行的原始状态以撤销删除。
- **提交删除操作时** 受影响的行并不总是从内存中删除。如果该行从未进行过同步，则该行会被删除。如果该行进行过同步，则该行不会被删除，因为删除操作首先需要与统一数据库同步。在下一次同步之后，该行将从内存中删除。
- **更新行时** 创建该行的新版本。旧行和新行的状态会设置为使旧行不再可见，而新行可见。
- **提交行更新操作时** 提交事务时，会修改该事务影响到的所有行的状态，以反映该提交操作。当同步更新时，旧行和新行均需要用于冲突检测和解决。然后旧行从数据库中删除，新行只是变成正常行。
- **添加行时** 行添加到数据库并标记为未提交。
- **提交添加行操作时** 行标记为已提交，同时也标记为需要与统一数据库同步。

### 另请参见

- “备份和恢复 UltraLite 中的数据”一节第 13 页
- “刷新单个或分组的事务”一节第 110 页
- “UltraLite 事务处理”一节第 14 页

## 校验 UltraLite 数据库

建议在数据库中无其它活动时进行校验。

可使用以下任一方法校验 UltraLite 数据库：

- Sybase Central 中的 [\[校验数据库向导\]](#)。
- 调用 ValidateDatabase 函数。
- ulvalid 实用程序。
- ValidateDatabase 方法。

#### 小心

应在没有任何连接对数据库进行更改时执行数据库校验；否则，可能会报告指出数据库损坏的错误，即使实际上并没有任何损坏。

#### ◆ 验证数据库 (Sybase Central)

1. 在 Sybase Central 的左窗格中，选择 UltraLite 数据库。
2. 选择 [\[文件\]](#) » [\[校验数据库\]](#)。
3. 请按照 [\[校验数据库向导\]](#) 中的说明进行操作。

#### 提示

您也可以通过下列任意一种方法从 Sybase Central 访问 [\[校验数据库向导\]](#)：

- 右击 UltraLite 数据库并选择 [\[校验数据库\]](#)。
- 选择 UltraLite 数据库，然后选择 [\[工具\]](#) » [\[UltraLite 11\]](#) » [\[校验数据库\]](#)。

#### 另请参见

- [“ValidateDatabase 方法”一节 《UltraLite - .NET 编程》](#)
- [“ValidateDatabase 函数”一节 《UltraLite - C 及 C++ 编程》](#)
- [“UltraLite 校验数据库实用程序 \(ulvalid\)”一节第 265 页](#)

## 备份和恢复 UltraLite 中的数据

如果一个使用 UltraLite 数据库的应用程序意外地停止，则 UltraLite 数据库会在该应用程序重新启动时自动恢复到一致状态。意外故障发生前所有刷新到内存的已提交事务均出现在 UltraLite 数据库中。在故障发生时未刷新的所有事务都将回退。

UltraLite 不使用事务日志来执行恢复。而是存储每一行的状态信息来确定在恢复时如何处理行。请参见 [“UltraLite 行状态”一节第 12 页](#)。

#### 备份

UltraLite 提供的保护功能是针对系统故障的，而不是针对介质故障的。制作 UltraLite 应用程序备份的最佳方法是与统一数据库进行同步。要恢复 UltraLite 数据库，可先建立一个空数据库，然后通过同步从统一数据库填充该数据库。

在较小的 UltraLite 部署中，可以将数据库文件复制到桌面计算机以提供手动备份。

**注意**

在 Palm OS 上，也可以将 PALM\_ALLOW\_BACKUP 连接参数设置为 true。此设置允许通过 HotSync 备份数据库。请参见“UltraLite PALM\_ALLOW\_BACKUP 连接参数”一节第 220 页。

**另请参见**

- “刷新单个或分组的事务”一节第 110 页

## UltraLite 事务处理

事务是自动执行操作的逻辑组：或者事务中的所有操作都存储在数据库中，或者一个都不存储。UltraLite 应用程序对 UltraLite 运行时的访问进行了序列化。当可能同时打开多个事务时，UltraLite 每次仅处理一个事务。此行为意味着应用程序无法：

- 拥有阻塞的事务（也称为死锁）。UltraLite 从不阻塞基于现有行锁的请求。这种情况下，UltraLite 立即返回一个错误。
- 覆盖未完成的修改。一个事务无法覆盖另一个事务的未完成的修改。当某个事务更改一行时，UltraLite 锁定该行直至提交或回滚该事务。锁定可以防止其它事务更改此行，但它们仍能读取此行。

例如，A 和 B 两个应用程序从数据库读取相同的行，并根据它们读取的数据为该行的某列计算新值。如果 A 使用它的新值更新行，然后 B 尝试修改同一行，则 B 会得到一个错误。尝试更改锁定的行会导致错误 SQLCODE SQLE\_LOCKED，而尝试更改已删除的行会导致错误 SQLCODE SQLCODE NOTFOUND。因此，应这样编写应用程序使它在尝试修改数据之后检查 SQLCODE 值。

有关如何处理错误的详细信息，请参见：

- UltraLite for C/C++：“处理错误”一节《UltraLite - C 及 C++ 编程》
- UltraLite.NET：“处理错误”一节《UltraLite - .NET 编程》
- UltraLite for M-Business Anywhere：“处理错误”一节《UltraLite - M-Business Anywhere 编程》

**编程提示**

所有的 UltraLite API—除了 C++ API—都可以在自动提交模式中操作。一些 API 在缺省情况下使用自动提交。

此缺省设置意味着在每次操作后，将自动提交每个事务。如果使用这些接口，则必须关闭自动提交设置，以便利用多操作事务。关闭自动提交的方法取决于所用的编程接口。在大多数接口中，自动提交是连接对象的一个属性。

请参见：

- UltraLite.NET：“管理事务”一节《UltraLite - .NET 编程》
- UltraLite for M-Business Anywhere：“管理事务”一节《UltraLite - M-Business Anywhere 编程》



## UltraLite 隔离级别

隔离级别定义一个事务中的操作对其它并发事务中的操作的可见程度。UltraLite 在自动提交模式中使用连接的缺省隔离级别 `read-committed`。对于 .NET，`read-committed` 是通过调用无参数 `ULConnection.BeginTransaction` 创建的新事务的缺省隔离级别。缺省 UltraLite 隔离级别提供最佳性能并确保数据的一致性。

对于 `ReadCommitted` 隔离级别：

- 阻止进行脏读
- 未应用读锁定
- 可以看到未提交的删除
- 允许非重复读取和幻像行
- 无法确保数据在事务执行过程中不发生更改

对于 `ReadUncommitted` 隔离级别：

- 允许进行脏读
- 未应用读锁定
- 允许非重复读取和幻像行
- 无法确保并发事务将不会修改行或回退对行所做的更改

您可以从 `ReadCommitted` 到 `ReadUncommitted` 更改隔离级别。对于 UltraLite C++，使用 `SetDatabaseOption` 方法更改隔离级别。对于 UltraLite.NET 2.0，调用 `ULConnection.BeginTransaction` 来创建具有 `ReadUncommitted` 隔离级别的事务。

### 注意

对于 .NET，不建议在事务处于激活状态时执行 `SetDatabaseOption`。它更改连接的隔离级别，但是不更新 `ULTransaction.IsolationLevel`。

当事务正在进行中时，不要使用 `SetDatabaseOption` 更改隔离；可能发生不可预期的结果。

## 隔离级别的副作用

因为缺省情况下，UltraLite 在 0 隔离级别下操作，所以可能出现以下副作用：

- 执行 `SELECT` 语句时，不需要任何锁定操作。
- 应用程序可以访问未提交的数据（也称为脏读）。在这种情况下，应用程序可以访问数据库中未提交的行，并且仍可以通过另一个事务获得回退。这种现象可以导致幻像行（在原始查询后添加的行，它们使在重复的查询中返回的结果集不同）。

有关演示脏读影响的教程，请参见“教程：脏读”一节《SQL Anywhere 服务器 - SQL 的用法》。有关演示幻像行的教程，请参见“教程：幻像行”一节《SQL Anywhere 服务器 - SQL 的用法》。

- 应用程序可以执行非可重复的读取。在这种情况下，应用程序从数据库中读取行，然后继续执行其它操作。然后，第二个应用程序更新/删除该行并提交更改。如果第一个应用程序尝试重新读取原始行，它将收到已更新的信息或发现原始行已被删除。

有关演示非可重复读取的影响的教程，请参见“教程：非可重复读取”一节《SQL Anywhere 服务器 - SQL 的用法》。

### 示例

请假设有两个连接（A 和 B），每个连接都有各自的事务。

1. 当连接 A 使用查询的结果集时，UltraLite 将当前行的副本读取到缓冲区中。

**注意**

读取某行时不会锁定该行。如果连接 A 读取某行但不对其进行修改，则连接 B 仍可以修改此行。

2. 当 A 修改当前行时，它更改的是缓冲区中的副本。当连接 A 调用 Update 方法或关闭结果集时，缓冲区中的副本就会写回到数据库。
3. 在该行上会放置一个写锁定以防止其它事务对其进行修改。连接 A 执行提交前，不会提交此修改。
4. 根据所做的修改，如果连接 B 读取当前行，可能会遇到以下情况：

连接 A 的修改	结果 <sup>1</sup>
行已被删除。	连接 B 在结果集中获取下一行。
行已被修改。	连接 B 获取行的最新副本。

<sup>1</sup> 连接 A 和 B 使用的查询不包含临时表。临时表可能导致其它副作用。

### 另请参见

- “BeginTransaction() 方法”一节《UltraLite - .NET 编程》
- “BeginTransaction(IsolationLevel) 方法”一节《UltraLite - .NET 编程》
- “SetDatabaseOption 方法”一节《UltraLite - .NET 编程》

---

# 实现 UltraLite 解决方案

## 目录

UltraLite 支持的平台和协议 .....	18
选择 UltraLite 数据管理组件 .....	19
选择 UltraLite 编程接口 .....	20

---

实现 UltraLite 解决方案时，请考虑以下事项：

- 有多少个应用程序需要连接到 UltraLite 数据库？并发连接的数量影响您需要 UltraLite 进程中运行时还是 UltraLite 引擎。要了解它们有何不同，请参见“[选择 UltraLite 数据管理组件](#)”一节第 19 页。
- 您想要支持何种平台？此选择可能会对您可使用哪些 API 来编写应用程序产生影响。请参见“[选择 UltraLite 编程接口](#)”一节第 20 页。
- 数据库会运行在哪些平台上？因为文件格式已统一，所以您可能会创建一个可在多个平台上运行的数据库。请参见“[创建和配置 UltraLite 数据库](#)”第 25 页。

### 提示

如果您需要创建一个适合多个平台的文件格式，则请使用 Sybase Central 中的 [[创建数据库向导](#)] 来帮助您确定单个数据库是否可行。

## UltraLite 支持的平台和协议

您可以使用中央统一数据库，通过 TCP/IP、HTTP 或 HTTPS 网络协议同步 UltraLite 数据库中的数据。

有关哪些平台支持不同的网络协议（也称为流）的详细信息，请参见 <http://www.sybase.com/detail?id=1062623>。

### 另请参见

- “UltraLite 客户端” 第 115 页
- “UltraLite 同步流的网络协议选项” 一节第 161 页
- “UltraLite 同步实用程序 (ulsync)” 一节第 255 页

## 选择 UltraLite 数据管理组件

UltraLite 允许您构建一个微型关系数据库解决方案，而无需建立单独数据库服务器的额外开销。相反，UltraLite 编程接口使用以下两种库类型之一。

### UltraLite 进程中运行时库

在 UltraLite 中，运行时和应用程序是同一过程中的一部分，这使得数据库因应用程序而异。对于所有的平台，运行时管理 UltraLite 数据库和内置同步操作。UltraLite 运行时无论何时最多可以管理 14 个数据库。请注意，在任何给定时间，只能有一个应用程序可以访问某个数据库。

链接到运行库需要使用不同于引擎的特殊导入库/DLL 对。有关详细信息，请参见“[编译和链接应用程序](#)”一节《[UltraLite - C 及 C++ 编程](#)》。如果需要启用了 TLS 的同步，则您还需要附加库。请参见“[通过启用 TLS 的同步部署 UltraLite](#)”一节第 51 页。

### UltraLite 数据库引擎客户端库

UltraLite 引擎仅适用于 Windows 桌面操作系统平台和 Windows Mobile 平台。引擎是可单独执行的，这提供了并发数据库/应用程序连接。每个应用程序在使用 UltraLite 引擎时都必须使用客户端库。此客户端库使每个应用程序能够与 UltraLite 引擎通信。UltraLite 引擎需要的系统资源比 UltraLite 运行时需要的多，因而可能降低性能。

链接到引擎需要使用不同于运行库的导入库/DLL 对。如果需要启用了 TLS 的同步或 AES FIPS 数据库加密，则您还需要附加库。请参见“[通过 AES\\_FIPS 数据库加密部署 UltraLite](#)”一节第 50 页和“[通过启用 TLS 的同步部署 UltraLite](#)”一节第 51 页。

#### 注意

请确保没有超过 UltraLite 引擎所支持的并发连接限制。

### 另请参见

- “[UltraLite 引擎实用程序 \(uleng11\)](#)”一节第 240 页
- “[UltraLite 中的并发](#)”一节第 11 页
- “[UltraLite 功能比较](#)”一节第 2 页
- “[UltraLite 限制](#)”一节第 6 页

## 选择 UltraLite 编程接口

所有 UltraLite API 公开核心数据库功能。以下的一些 API 可与开发环境集成以简化编程任务：

- C/C++ 接口
- 用于 C/C++ 的嵌入式 SQL
- UltraLite.NET (C# 或 VB.NET)
- M-Business Anywhere (JavaScript)

UltraLite API 提供了不同的数据访问模式，其中包括一个简单的、基于表的数据访问接口，并为更复杂的查询提供了动态 SQL。结合这些优点，UltraLite 为应用程序开发人员提供了灵活的体系结构，允许创建用于不同部署环境的应用程序。

### 注意

此版本的 UltraLite 不支持 Pocket Builder。Sybase PocketBuilder 不随 SQL Anywhere 提供。有关详细信息，请与 Sybase 联系 (<http://www.sybase.com/products/developmentintegration/pocketbuilder>)。

### 另请参见

- UltraLite.NET: [UltraLite - .NET 编程](#)
- UltraLite C/C++ 和嵌入式 SQL: [UltraLite - C 及 C++ 编程](#)
- UltraLite for M-Business Anywhere: [UltraLite - M-Business Anywhere 编程](#)

### ◆ 选择编程接口

1. 选择目标平台。UltraLite 支持 Palm OS、Windows Mobile、Windows XP/嵌入式 Windows XP 和 Java OS。
2. 对于您需要支持的每一个平台，确定 API 是否支持该平台。不同的 API 支持不同的平台。如果您正在进行跨平台开发，请选择支持您想要的所有目标的 API。

使用此支持总表可帮助您快速标识开发项目。

部署目标	UltraLite for C/C++ 和嵌入式 SQL	UltraLite.NET <sup>1</sup>	UltraLite for M-Business Anywhere <sup>2</sup>
Palm OS	版本 4 +	N/A	版本 5.0 +
Windows Mobile	CE 5.0 +	CE 5.0 和 .NET compact framework 2.0	版本 3.0 +
嵌入式 Windows XP	支持	.NET framework 2.0	版本 5.0 +

部署目标	UltraLite for C/C++ 和嵌入式 SQL	UltraLite.NET <sup>1</sup>	UltraLite for M-Business Anywhere <sup>2</sup>
Java	Java SE (1.5 版及更高版本) RIM BlackBerry OS (4.1 版及更高版本)	N/A	N/A

<sup>1</sup> 开发为 Microsoft Visual Studio.NET 的扩展。该驱动程序支持 ADO.NET 版本 2.0。

<sup>2</sup> 用于 JavaScript 中 UltraLite 编程的基于浏览器的部署。

3. 请考虑以下要求的影响，然后完成您的选择：

**SQL Anywhere 兼容性** 如果您关心数据库与 SQL Anywhere 之间的兼容性，请考虑以下内容：

- SQL Anywhere 嵌入式 SQL 支持为 UltraLite 和 SQL Anywhere 数据库提供了一种常用的编程接口。
- ADO.NET 提供了在 UltraLite 组件和 SQL Anywhere 之间共享的常用编程模型。

在诸如 Windows Mobile（在该平台上两种数据库都可用）的平台上维护常用接口可能非常有用。如果您需要将应用程序从 UltraLite 移动到 SQL Anywhere 数据库，您应使用嵌入式 SQL 或 ADO.NET 以使应用程序迁移更加容易。

**提示**

即使存在常用的接口，好的开发实践也会要求您在编写应用程序时创建抽象的数据访问层。

**简化部署** 如果简化 UltraLite 部署是一个问题，请考虑使用 M-Business Anywhere API 编程。您的最终用户可以同时下载 UltraLite 应用程序和数据库。

**应用程序大小** 如果创建占用资源空间最少的应用程序是优先考虑的事情，则您应使用 C/C++ API 编写应用程序。这些应用程序通常都会获得最佳性能，且仍然保持着很小的应用程序文件大小。

**应用程序性能** 每个 API 会产生不同的性能结果。例如，UltraLite.NET 的性能可能不如 UltraLite C++ 优良，而 UltraLite 嵌入式 SQL 的性能可能比 UltraLite C++ 更佳。

---



# 使用 UltraLite 数据库

本节介绍如何创建和使用 UltraLite 数据库。

---

创建和配置 UltraLite 数据库 .....	25
连接到 UltraLite 数据库 .....	37
将 UltraLite 部署到设备 .....	47
使用 UltraLite 数据库 .....	61
UltraLite CustDB 示例 .....	85
UltraLite 性能与优化 .....	99



---

# 创建和配置 UltraLite 数据库

## 目录

创建 UltraLite 数据库 .....	26
为 UltraLite 选择数据库创建参数 .....	30
UltraLite 字符集 .....	32
保护 UltraLite 数据库 .....	35

---

数据库创建方法有两种类型：

- 桌面操作系统创建方法，使用为此目的而设计的 UltraLite 管理工具。
- 设备上的创建方法，使用 UltraLite API。设备上的创建方法主要在各个特定于 API 的 UltraLite 编程书籍中进行介绍。

一旦创建了数据库，便可连接到该数据库并构建表和其它数据库对象。

如果您不想使用缺省值，则必须在创建数据库时设置创建参数值。这些值不能在创建数据库后更改。如果需要更改任何数据库属性，则必须重新创建数据库，并为创建参数指定不同的值。

## 在多个平台之间共享数据库

在不同操作系统所施加的配置差异范围内，您可能能够将数据库从一个设备复制到另一个设备。如果您对多个平台之间的属性兼容性不确定，则应在 Sybase Central 中使用 **[UltraLite 创建数据库向导]** 创建数据库。此向导会帮助您处理文件兼容性逻辑。这样做可以防止创建部署设备组合上不支持的文件。

## 另请参见

- “UltraLite 优化策略” 一节第 112 页
- “为 UltraLite 选择数据库创建参数” 一节第 30 页
- “变更 UltraLite 持久性数据库选项设置” 一节第 206 页
- “使用 UltraLite 数据库” 第 61 页

## 创建 UltraLite 数据库

**UltraLite 创建数据库向导** 如果您在浏览可用数据库创建参数方面需要帮助，请选择此方法。此向导会根据所选择的目标平台对您可以配置的内容实施限制，从而简化您的选择。创建了数据库后，即会显示命令行语法，您可以将其记录下来并在 `ulcreate` 实用程序中使用。请参见“[使用 \[创建数据库向导\] 创建数据库](#)”一节第 26 页。

**MobiLink 创建同步模型向导** 如果您要创建具有远程 UltraLite 数据库和一个集中式统一数据库的同步系统，请选择此方法。

请参见“[从 MobiLink 同步模型创建 UltraLite 数据库](#)”一节第 27 页。

**命令行** 选择以下任一实用程序：

- 如果您对数据库创建参数非常熟悉并且想要使用一种快速替代方法来创建新的空 UltraLite 数据库，请使用 `ulcreate` 实用程序。此选项对于以批处理操作创建数据库尤为有用。请参见“[从命令提示符创建 UltraLite 数据库](#)”一节第 26 页。
- 如果您要创建一个新的空 UltraLite 数据库且该数据库源自于 SQL Anywhere 参考数据库模式，请使用 `ulinit` 实用程序。请参见“[从 SQL Anywhere 参考数据库创建 UltraLite 数据库](#)”一节第 27 页。
- 如果您将以某个 XML 文件充当新 UltraLite 数据库的模式和/或数据源点，请使用 `ulload` 实用程序。请参见“[从 XML 创建 UltraLite 数据库](#)”一节第 28 页。

## 使用 [创建数据库向导] 创建数据库

可以在 Sybase Central 中使用 [\[创建数据库向导\]](#) 创建数据库。

### ◆ 创建新的 UltraLite 数据库 (Sybase Central)

1. 选择 [\[开始\]](#) » [\[程序\]](#) » [\[SQL Anywhere 11\]](#) » [\[Sybase Central\]](#)。
2. 选择 [\[工具\]](#) » [\[UltraLite 11\]](#) » [\[创建数据库\]](#)。
3. 请按照 [\[创建数据库向导\]](#) 中的说明进行操作。

## 从命令提示符创建 UltraLite 数据库

可以从命令提示符使用 `ulcreate` 实用程序创建数据库。在此实用程序中，可以加入实用程序选项来配置数据库。

### 创建新的 UltraLite 数据库（命令提示符）

运行 `ulcreate` 实用程序，在其中加入任何必要的选项。例如，要创建名为 `test.udb` 的区分大小写的 UTF-8 数据库，如果该数据库已存在，将其覆盖，运行以下命令：

```
ulcreate -c "DBF=test.udb" -o "case=respect;utf_encoding=1" -y
```

在连接字符串中提供数据库文件是在其它选项之后指定数据库文件的一种替代方法。例如：

```
ulcreate -o "case=respect;utf_encoding=1" -y test.udb
```

## 从 MobiLink 同步模型创建 UltraLite 数据库

要简化开发，MobiLink 需包含 [\[创建同步模型向导\]](#)，该向导可用于创建 UltraLite 数据库和服务器端同步逻辑。

一旦创建了模型，便可在 Sybase Central 中工作于 [\[MobiLink 模型\]](#) 模式下，并在部署同步模型之前对其进行自定义。模型准备好之后，就可以部署该模型以生成同步应用程序所需的脚本和表。

请参见 [“MobiLink 模型”](#) 《[MobiLink - 入门](#)》。

## 从 SQL Anywhere 参考数据库创建 UltraLite 数据库

参考数据库是一个 SQL Anywhere 数据库，它用作要创建的 UltraLite 数据库的模板。UltraLite 数据库是此参考数据库中的列、表和索引的子集。在参考数据库中选择这些对象作为发布内容的一部分。

如果您要首先使用体系结构工具（如 Sybase PowerDesigner）进行数据建模，从参考数据库创建数据库可能会有用。

用从参考数据库创建数据库，请实用 ulimit 实用程序。

### ◆ 从参考数据库初始化/抽取新 UltraLite 数据库（命令提示符）

#### 1. 创建新 SQL Anywhere 数据库作为参考数据库。

使用 dbinit 实用程序或 Sybase Central 都可以创建新 SQL Anywhere 数据库。还可以从非 SQL Anywhere 数据库创建 SQL Anywhere 数据库，方法是从这些第三方文件中迁移数据。

请参见 [“创建数据库”](#) 一节 《[SQL Anywhere 服务器 - 数据库管理](#)》。

#### 配置数据库时注意 UltraLite 的使用

生成 UltraLite 数据库时使用的设置与参考数据库中的设置相同。通过在参考数据库中设置以下选项，还可以控制 UltraLite 数据库的行为：

- 日期格式
- 日期顺序
- 最接近的世纪值
- 精度
- 小数位数
- 时间格式
- 时间戳格式

#### 2. 通过添加 UltraLite 数据库所需的对象来准备参考数据库：

- **表和键** 添加表并记得设置主键，因为 UltraLite 需要主键。如果需要，还可以指派在 UltraLite 应用程序内所需的外键关系。可以使用 Sybase Central、Sybase PowerDesigner

Physical Data Model 或其它数据库设计工具。请参见“使用 UltraLite 表和列”一节第 62 页。

- **索引** 索引可以显著地提高性能，特别是在速度较慢的设备上。请注意，主键会自动建立索引，而其它列则不然。请参见“何时使用索引”一节第 71 页。
- **发布** 如果要在不同时间同步不同的表，请使用发布。可以使用多个 UltraLite 发布来定义表子集并对它们设置同步优先级。请参见“UltraLite 中的发布”一节第 124 页。

### 性能提示

如果 UltraLite 应用程序经常以特定顺序检索信息，请考虑专门为此目的向参考数据库添加索引。请参见“使用索引扫描”一节第 100 页。

3. 运行 `ulinit` 实用程序，在其中加入任何必要的选项。

例如，要初始一个名为 `customer.udb` 的 UltraLite 数据库并使表包含在两个不同的发布中，请运行以下命令。Pub1 可以包含一个小的表子集来进行优先级同步，而 Pub2 可以包含大量数据。

```
ulinit -a DBF=MySource.db -c DBF=customer.udb -n Pub1 -n Pub2
```

### 另请参见

- “使用 [创建数据库向导] 创建数据库”一节第 26 页
- “从命令提示符创建 UltraLite 数据库”一节第 26 页
- “从 XML 创建 UltraLite 数据库”一节第 28 页
- “UltraLite 初始化数据库实用程序 (ulinit)”一节第 249 页
- “为 UltraLite 选择数据库创建参数”一节第 30 页
- “升级 UltraLite”一节 《SQL Anywhere 11 - 更改和升级》

## 从 XML 创建 UltraLite 数据库

可以使用 XML 作为管理 UltraLite 数据库的中间格式，只要该格式符合 UltraLite 使用要求。XML 的使用方法如下：

- 使用不同的一组数据库属性/选项将数据装载到新数据库中。
- 从先前版本的 UltraLite 所创建的数据库升级模式。
- 创建一个可签入版本控制系统的 UltraLite 数据库文本版本。

UltraLite 不能使用任意的 XML 文件。`install-dir\UltraLite` 目录包含一个 `usm.xsd` 文件。使用此文件查看 XML 格式。

### ◆ 从 XML 文件创建 UltraLite 数据库

1. 将 XML 文件保存到所选择的目录。您可以：

- 将数据库导出/卸载到某个 XML 文件。如果要卸载 SQL Anywhere 数据库，请使用任何一种支持的导出方法。请参见“以 XML 形式导出关系数据”一节 《SQL Anywhere 服务器 - SQL 的用法》。

- 从另一个来源取得 XML 输出—该来源可以是另一个关系数据库，甚至可以是用于将事务记录到文件的 Web 站点。务必要确保 XML 的格式始终满足 UltraLite 要求。
2. 运行 ulload 实用程序，在其中加入任何必要的选项。

例如，在文件 *sample.udb* 中通过表格式和 *sample.xml* 中的数据创建新 UltraLite 数据库：

```
ulload -c DBF=sample.udb sample.xml
```

#### 另请参见

- [“使用 \[创建数据库向导\] 创建数据库”一节第 26 页](#)
- [“从命令提示符创建 UltraLite 数据库”一节第 26 页](#)
- [“从 SQL Anywhere 参考数据库创建 UltraLite 数据库”一节第 27 页](#)
- [“升级 UltraLite”一节 《SQL Anywhere 11 - 更改和升级》](#)
- [“UltraLite 装载 XML 到数据库实用程序 \(ulload\)”一节第 252 页](#)
- [“为 UltraLite 选择数据库创建参数”一节第 30 页](#)

## 在第一次连接时创建 UltraLite 数据库

如果在连接时无法检测到数据库，可对应用程序进行编程来创建一个新的 UltraLite 数据库。然后，该应用程序即可使用 SQL 创建表、索引、外键等。要填充数据库，请与统一数据库同步。

#### 注意事项

添加附加数据库创建和 SQL 代码之后，应用程序大小会显著增长。不过，此选项可以简化部署，因为您只需要将应用程序部署到设备。在一些生产前开发周期中，可能还需要在设备上删除并重新构造供测试用的数据库。

#### 另请参见

- [“创建 UltraLite 数据库”一节第 26 页](#)
- UltraLite for C/C++: [“CreateDatabase 函数”一节 《UltraLite - C 及 C++ 编程》](#)
- UltraLite.NET: [“CreateDatabase 方法”一节 《UltraLite - .NET 编程》](#)
- UltraLite for M-Business Anywhere: [“createDatabase 方法”一节 《UltraLite - M-Business Anywhere 编程》](#)

## 为 UltraLite 选择数据库创建参数

UltraLite 数据库创建参数在组成数据库模式的系统表中被记录为 *name=value* 对。通过将这些创建参数存储在系统表中，用户和/或应用程序能够像访问其它数据一样访问它们。请参见“[sysuldata 系统表](#)”一节第 276 页。

### 访问创建参数值

创建了数据库之后，便无法更改创建参数值。不过，可以在 Sybase Central 中查看相应的数据库属性。请参见“[访问 UltraLite 数据库属性](#)”一节第 199 页。

还可以通过调用适合于 API 的 `GetDatabaseProperty` 函数从 UltraLite 应用程序以编程方式访问数据库属性。

有关 API 特定的详细信息，请参见：

- UltraLite C/C++: “[GetDatabaseProperty 函数](#)”一节《[UltraLite - C 及 C++ 编程](#)》
- UltraLite.NET: “[GetDatabaseProperty 方法](#)”一节《[UltraLite - .NET 编程](#)》
- M-Business: “[getDatabaseProperty 方法](#)”一节《[UltraLite - M-Business Anywhere 编程](#)》

除了这些数据库创建参数之外，还可以使用数据库选项和连接参数进一步配置数据库的其它方面。请参见：

- “[UltraLite 数据库选项](#)”第 201 页
- “[连接到 UltraLite 数据库](#)”第 37 页
- “[UltraLite 连接参数](#)”第 207 页

属性	说明
case	将 UltraLite 数据库中的字符串比较设置成区分大小写。请参见“ <a href="#">UltraLite 大小写创建参数</a> ”一节第 167 页。
checksum_level	设置数据库中校验和校验的级别。请参见“ <a href="#">UltraLite checksum_level 创建参数</a> ”一节第 168 页。
collation	设置 UltraLite 数据库使用的归类序列。是否与 UTF-8 属性一起设置此属性将决定数据库的字符集。请参见“ <a href="#">UltraLite 字符集</a> ”一节第 32 页和“ <a href="#">UltraLite 归类创建参数</a> ”一节第 170 页和“ <a href="#">UltraLite utf8_encoding 创建参数</a> ”一节第 193 页。
date_format	设置从数据库中检索日期所用的缺省字符串格式。请参见“ <a href="#">UltraLite date_format 创建参数</a> ”一节第 171 页。
date_order	控制对年、月、日的日期顺序的解释。请参见“ <a href="#">UltraLite date_order 创建参数</a> ”一节第 174 页。
fips	通过使用 Certicom 认证的加密算法，控制符合 AES FIPS 的数据加密。FIPS 编码是一种高度加密方式。请参见“ <a href="#">保护 UltraLite 数据库</a> ”一节第 35 页和“ <a href="#">UltraLite fips 创建参数</a> ”一节第 176 页。



属性	说明
max_hash_size	以字节为单位设置缺省索引散列大小。请参见“ <a href="#">UltraLite max_hash_size 创建参数</a> ”一节第 177 页。
nearest_century	控制字符串到日期的转换中对两位的年份的解释。请参见“ <a href="#">UltraLite nearest_century 创建参数</a> ”一节第 179 页。
obfuscate	控制是否对数据库中的数据进行模糊处理。模糊处理是一种简单的加密方式。请参见“ <a href="#">保护 UltraLite 数据库</a> ”一节第 35 页和“ <a href="#">UltraLite obfuscate 创建参数</a> ”一节第 181 页。
page_size	定义数据库页面大小。请参见“ <a href="#">UltraLite page_size 创建参数</a> ”一节第 182 页。
precision	指定小数点算术结果中的最多位数。请参见“ <a href="#">UltraLite 精度创建参数</a> ”一节第 184 页。
scale	指定算术结果被截断为最大精度值时小数点后的最小位数。请参见“ <a href="#">UltraLite 小数位数创建参数</a> ”一节第 185 页。
time_format	为从数据库中检索得到的时间设置格式。请参见“ <a href="#">UltraLite time_format 创建参数</a> ”一节第 187 页。
timestamp_format	为从数据库检索得到的时间戳设置格式。请参见“ <a href="#">UltraLite timestamp_format 创建参数</a> ”一节第 189 页。
timestamp_increment	确定 UltraLite 中时间戳的截断方式。请参见“ <a href="#">UltraLite timestamp_increment 创建参数</a> ”一节第 192 页。
utf8_encoding	使用 UTF-8 格式（Unicode 的 8 位多字节编码）对数据进行编码。请参见“ <a href="#">UltraLite 字符集</a> ”一节第 32 页和“ <a href="#">UltraLite utf8_encoding 创建参数</a> ”一节第 193 页。

## UltraLite 字符集

字符串的比较结果和排序顺序部分取决于数据库的字符集、归类和编码属性。

为数据库选择正确的字符集、归类和编码属性主要由以下因素决定：

- 所需的排序顺序。一般来说，应选择可对要存储在数据库中的字符进行最佳排序的归类。
- 设备的平台。所支持设备之间的要求可能会有所变化，一些设备要求使用 UTF-8 进行字符编码。如果需要使用多个设备，则需要确定数据库是否可以共享。
- 如果要同步数据，需确定统一数据库支持哪些语言和字符集。必须确保 UltraLite 数据库和统一数据库中使用的字符集兼容。否则，如果一个数据库字符集中的字符在另一个数据库字符集中不存在，则数据可能会丢失或发生意想不到的变化。如果已在多语言环境中部署了 UltraLite，还应使用 UTF-8 对 UltraLite 数据库进行编码。

同步时，MobiLink 服务器会按如下方式转换字符：

1. 将 UltraLite 数据库字符转换为 Unicode。
2. 将 Unicode 字符转换为统一数据库的字符集。

### 另请参见

- “字符集编码的 UltraLite 平台要求” 一节第 33 页
- “UltraLite 归类创建参数” 一节第 170 页
- “UltraLite utf8\_encoding 创建参数” 一节第 193 页
- “字符集、编码和归类概述” 一节 《SQL Anywhere 服务器 - 数据库管理》
- “UltraLite 连接参数” 第 207 页
- “字符集注意事项” 《MobiLink - 服务器管理》
- “UltraLite 大小写创建参数” 一节第 167 页
- “保护 UltraLite 数据库” 一节第 35 页

## 更改字符集和归类

一旦 UltraLite 写入文件，便无法更改数据库的字符集、归类和编码属性。必须重新创建数据库。

### ◆ 重新创建具有新归类的 UltraLite 数据库

1. 卸载数据库。

可以使用 Sybase Central 或 ulunload 实用程序中的 [卸载数据库向导]。请参见 “UltraLite 卸载旧数据库实用程序 (ulunload)” 一节第 260 页。

2. 创建具有适当归类的新数据库。
3. 可选。如果需要将转换数据以匹配新的归类。
4. 重装数据库。

可以使用 Sybase Central 或 ulload 实用程序中的 [装载数据库向导]。请参见 “UltraLite 装载 XML 到数据库实用程序 (ulload)” 一节第 252 页。

## 字符集编码的 UltraLite 平台要求

每个平台都有特定的字符集和编码要求。

### Palm OS

决不要使用 UTF-8 编码；Palm 不支持 Unicode 字符。务必选择与预定设备的代码页相匹配的归类。

### Windows 桌面操作系统和 Windows Mobile

在 Windows 上使用 UTF-8 编码的数据库时，应向数据库传递宽字符。如果在这些平台上使用 UTF-8 编码，则 UltraLite 会认为非宽字符串参数是 UTF-8 编码，而 UTF-8 编码不是 Windows 上本来使用的字符集。连接字符串是个例外，此时会认为字符串参数在活动代码页中。但是，通过使用宽字符，可以避免这种复杂化。

### 另请参见

- [“UltraLite utf8\\_encoding 创建参数”一节第 193 页](#)
- [“字符集、编码和归类概述”一节《SQL Anywhere 服务器 - 数据库管理》](#)
- [“UltraLite 连接参数”第 207 页](#)
- [“字符集注意事项”《MobiLink - 服务器管理》](#)
- [“保护 UltraLite 数据库”一节第 35 页](#)

## UltraLite 支持的归类

下表列出了 UltraLite 中支持的 CHAR 归类。也可以执行以下命令来生成该列表：

```
ulcreate -l
```

归类标签	说明
1250LATIN2	代码页 1250，Windows 拉丁语 2，中欧语言/东欧语言
1250POL	代码页 1250，Windows 拉丁语 2，波兰语
1251CYR	代码页 1251，Windows 西里尔语
1252LATIN1	代码页 1252，Windows 拉丁语 1，西方语言
1252NOR	代码页 1252，Windows 拉丁语 1，挪威语
1252SPA	代码页 1252，Windows 拉丁语 1，西班牙语
1252SWEFIN	代码页 1252，Windows 拉丁语 1，瑞典语/芬兰语
1253ELL	代码页 1253，Windows 希腊语，带扩展的 ISO8859-7
1254TRK	代码页 1254，Windows 土耳其语，带扩展的 ISO8859-9

归类标签	说明
1254TRKALT	代码页 1254, Windows 土耳其语, 带扩展的 ISO8859-9, I-dot 等同于 I-no-dot
1255HEB	代码页 1255, Windows 希伯来语, 带扩展的 ISO8859-8
1256ARA	代码页 1256, Windows 阿拉伯语, 带扩展的 ISO8859-6
1257LIT	代码页 1257, Windows 波罗的海, 立陶宛语
874THAIBIN	代码页 874, Windows 泰语, ISO8859-11, 二进制排序
932JPN	代码页 932, 带有 Microsoft 扩展的日语 Shift-JIS
936ZHO	代码页 936, 简体中文, PRC GBK
949KOR	代码页 949, 朝鲜语 KS C 5601-1987 编码, Wansung
950ZHO_HK	代码页 950, 繁体中文, 具有 HKSCS 的 Big 5 编码
950ZHO_TW	代码页 950, 繁体中文, Big 5 编码
EUC_CHINA	简体中文, GB 2312-80 编码
EUC_JAPAN	日语 EUC JIS X 0208-1990 和 JIS X 0212-1990 编码
EUC_KOREA	代码页 1361, 朝鲜语 KS C 5601-1992 8 位编码, Johab
EUC_TAIWAN	代码页 964, EUC-TW 编码
ISO1LATIN1	ISO8859-1, ISO 拉丁语 1, 西方语言, 拉丁语 1 排序
ISO9LATIN1	ISO8859-15, ISO 拉丁语 9, 西方语言, 拉丁语 1 排序
ISO_1	ISO8859-1, ISO 拉丁语 1, 西方语言
ISO_BINENG	二进制排序, 英语 ISO/ASCII 7 位字母大小写映射
UTF8BIN	UTF-8, Unicode 的 8 位多字节编码, 二进制排序

## 保护 UltraLite 数据库

缺省情况下，UltraLite 数据库在磁盘上没有经过加密。当使用诸如十六进制编辑器这样的查看工具时，可以看到文本和二进制列。如果为了提高安全性需要加密数据，请考虑以下选项：

- **模糊处理** 也称为简单加密，此选项用于防止对数据库中的数据进行不经意地访问。它提供的安全措施没有高度加密那么多。模糊处理对性能的影响最小。可使用 `obfuscate` 创建参数启用模糊处理。最终用户不需要提供相应的连接参数。不需要任何特殊配置就可以在设备上使用简单的模糊处理。请参见“[UltraLite obfuscate 创建参数](#)”一节第 181 页。
- **AES 128 位高度加密** 可使用 AES 128 位算法对 UltraLite 数据库进行高度加密，该算法与用来加密 SQL Anywhere 数据库的算法相同。为对付技术高超而又决意要非法获取数据访问权的人，高度加密确实可以提高安全性，但同时也会明显影响性能。在向导中设置加密的方法是选择 **[加密数据库]** 选项，然后选择 **[AES 高度加密]**。使用创建实用程序，可使用 `key` 连接参数设置密钥。创建了数据库之后，最终用户在连接到数据库时也会使用这个参数。不需要任何特殊配置就可以在设备上使用 AES 加密。请参见“[UltraLite fips 创建参数](#)”一节第 176 页。
- **AES FIPS 140-2 兼容加密** UltraLite 提供了与 FIPS 140-2 美国和加拿大政府标准兼容的加密库（使用经过 Certicom 认证的加密模块）。可使用 `fips` 创建参数设置 FIPS 兼容加密。用户必须提供他们连接字符串中所需的密匙。AES FIPS 加密要求对设备进行适当配置。请参见“[通过 AES\\_FIPS 数据库加密部署 UltraLite](#)”一节第 50 页和“[UltraLite fips 创建参数](#)”一节第 176 页。

### 提示

MobiLink 服务器的同步流可使用公共/专用密钥加密流数据。为便于部署，可以在创建 UltraLite 数据库时将证书嵌入其中。请参见“[配置 MobiLink 客户端以使用传送层安全](#)”一节《[SQL Anywhere 服务器 - 数据库管理](#)》。

### 注意

FIPS 和 AES 数据库加密类型都使用 128 位 AES。因此，如果使用相同的加密密钥，则不管选择哪个标准，数据库都会以相同的方式进行加密。

### 小心

创建了数据库之后可以更改加密密钥，但必须格外小心。请参见：

- UltraLite for C++: “[ChangeEncryptionKey 函数](#)”一节《[UltraLite - C 及 C++ 编程](#)》
- UltraLite.NET: “[ChangeEncryptionKey 方法](#)”一节《[UltraLite - .NET 编程](#)》
- UltraLite for M-Business Anywhere: “[changeEncryptionKey 方法](#)”一节《[UltraLite - M-Business Anywhere 编程](#)》

此操作代价很高且不可恢复：如果中途终止操作，数据库将会完全丢失。

对于高度加密的数据库，请务必将密钥的副本保存在安全的位置。如果丢失了加密密钥，就没有办法访问数据，即使有技术支持人员协助也不行。此时必须放弃该数据库并创建一个新的数据库。

### 另请参见

- [“UltraLite fips 创建参数”一节第 176 页](#)
- [“UltraLite obfuscate 创建参数”一节第 181 页](#)
- [“UltraLite DBKEY 连接参数”一节第 214 页](#)
- [“通过启用 TLS 的同步部署 UltraLite”一节第 51 页](#)
- UltraLite.NET: [“加密和模糊处理”一节《UltraLite - .NET 编程》](#)
- UltraLite for C++: [“对数据进行加密”一节《UltraLite - C 及 C++ 编程》](#)
- UltraLite for M-Business Anywhere: [“数据库加密和模糊处理”一节《UltraLite - M-Business Anywhere 编程》](#)

---

# 连接到 UltraLite 数据库

## 目录

UltraLite 数据库连接参数 .....	38
UltraLite 用户验证 .....	44
使用 ULSQLCONNECT 环境变量存储 UltraLite 参数 .....	46

---

在任何事务可能发生前，任何使用数据库的应用程序都必须建立与该数据库的连接。应用程序可以是 UltraLite 命令行实用程序、Sybase Central 工具或 Interactive SQL 的连接窗口，或您自己的自定义应用程序。

通过连接到 UltraLite 数据库，形成了应用程序的所有活动发生通过的通道。每个连接都尝试创建数据库特定的 SQL 事务。

## UltraLite 数据库连接参数

在创建时，配置的某些连接参数与定义的属性重叠。在创建时，通常定义所需的属性。在连接时，必须提供创建时配置的值。理解这些共享设置的差异和共性非常重要，以便您知道何时需要什么设置。

参数名	说明
CACHE_SIZE	定义数据库高速缓存的大小。请参见“ <a href="#">UltraLite CACHE_SIZE 连接参数</a> ”一节第 208 页。
CON	指定当前连接的名称。请参见“ <a href="#">UltraLite CON 连接参数</a> ”一节第 211 页。
DBF 和 CE_FILE、PALM_FILE 或 NT_FILE	<p>在创建时这些参数设置数据库的位置。对随后的连接，它们通知 UltraLite 在何处找到该文件。</p> <p>如果创建单个平台的应用程序或连接到 UltraLite 管理工具，可以使用 DBF。如果正在编制连接到不同特定平台数据库的 UltraLite 客户端程序，则请使用其它特定平台的版本。请参见：</p> <ul style="list-style-type: none"> <li>● “<a href="#">UltraLite DBF 连接参数</a>”一节第 212 页</li> <li>● “<a href="#">UltraLite CE_FILE 连接参数</a>”一节第 209 页</li> <li>● “<a href="#">UltraLite PALM_FILE 连接参数</a>”一节第 221 页</li> <li>● “<a href="#">UltraLite NT_FILE 连接参数</a>”一节第 218 页</li> </ul>
DBN	通过名称而不是文件名标识正在运行的数据库。请参见“ <a href="#">UltraLite DBN 连接参数</a> ”一节第 215 页。
DBKEY	在创建时，此参数设置要使用的加密密钥。对随后的连接，为数据库命名然后传递同样的加密密钥。如果命名了不正确的密钥，则连接失败。请参见“ <a href="#">UltraLite DBKEY 连接参数</a> ”一节第 214 页。
MIRROR_FILE	指定数据库镜像文件的名称。请参见“ <a href="#">UltraLite MIRROR_FILE 连接参数</a> ”一节第 216 页。
PALM_ALLOW_BACKUP	控制 Palm 设备上的 HotSync 上的备份行为。请参见“ <a href="#">UltraLite PALM_ALLOW_BACKUP 连接参数</a> ”一节第 220 页。
PWD	在创建时，为用户设置初始口令。对随后的连接，为用户 ID 提供口令。请参见“ <a href="#">UltraLite PWD 连接参数</a> ”一节第 223 页。
RESERVE_SIZE	预先分配 UltraLite 数据库所需的文件系统空间，而不实际插入任何数据。请参见“ <a href="#">UltraLite RESERVE_SIZE 连接参数</a> ”一节第 224 页。



参数名	说明
START	指定 UltraLite 引擎可执行文件的位置，然后启动它。请参见“ <a href="#">UltraLite START 连接参数</a> ”一节第 225 页。
UID	在创建时，设置初始用户 ID。对随后的连接，向数据库标识用户。用户 ID 必须是存储在 UltraLite 数据库中的最多四个用户 ID 中的一个。请参见“ <a href="#">UltraLite UID 连接参数</a> ”一节第 226 页。

### 另请参见

- “[解释用户 ID 和口令组合](#)”一节第 44 页
- “[UltraLite page\\_size 创建参数](#)”一节第 182 页

## 提供 UltraLite 连接参数

如何收集连接信息取决于要求输入的系统化或自动化的程度。输入越系统化，连接信息就越可靠。可通过不同的方法收集连接详细信息，这取决于是从自定义 UltraLite 应用程序还是从用于 UltraLite 的 SQL Anywhere 管理工具之一进行连接。

方法	管理工具	自定义应用程序
<p>如果需要将用户验证为四个所支持的数据库用户之一，会在连接时提示最终用户。UltraLite 图形管理工具使用连接对象。</p> <p>请尽可能使用 <code>ULConnectionParms</code> 或 <code>ConnectionParms</code> 对象。与使用作为 <code>Open</code> 方法参数的连接字符串相比，它可以提供更方便的检查和更系统的接口。请参见：</p> <ul style="list-style-type: none"> <li>● UltraLite.NET: “<a href="#">验证用户</a>”一节《<a href="#">UltraLite - .NET 编程</a>》</li> <li>● UltraLite C/C++: “<a href="#">验证用户</a>”一节《<a href="#">UltraLite - C 及 C++ 编程</a>》</li> <li>● UltraLite for M-Business Anywhere: “<a href="#">验证用户</a>”一节《<a href="#">UltraLite - M-Business Anywhere 编程</a>》</li> <li>● UltraLite for Embedded SQL: “<a href="#">验证用户</a>”一节《<a href="#">UltraLite - C 及 C++ 编程</a>》</li> </ul>	X	X

方法	管理工具	自定义应用程序
<p>如果不需要用户验证，则使用连接字符串。不验证用户的常见原因可能是因为部署是单用户设备，或很不方便在每次启动应用程序时都提示用户。如果需要连接到数据库，UltraLite 命令行实用程序通常使用连接字符串。也可以编制 UltraLite 应用程序从存储文件中读取值，或将其硬编码到应用程序中。请参见：</p> <ul style="list-style-type: none"> <li>● UltraLite.NET： “连接到数据库” 一节 《UltraLite - .NET 编程》</li> <li>● UltraLite C/C++： “连接到数据库” 一节 《UltraLite - C 及 C++ 编程》</li> <li>● UltraLite for M-Business Anywhere： “连接到 UltraLite 数据库” 一节 《UltraLite - M-Business Anywhere 编程》</li> <li>● UltraLite for embedded SQL： “连接到数据库” 一节 《UltraLite - C 及 C++ 编程》</li> </ul>	X <sup>1</sup>	X <sup>3</sup>
<p>如果想要存储您重复使用的连接参数，则使用 ULSQLCONNECT 环境变量。通过存储参数，您无需在开发阶段重复提供它们。在 ULSQLCONNECT 中作为参数提供的值将变为 UltraLite 桌面操作系统管理工具的缺省值。</p> <p>所有的 UltraLite 桌面操作系统管理工具将按照参数优先级规则，为连接字符串中任何未提供的缺少的参数检查 ULSQLCONNECT 值。要替换这些值，在连接字符串中提供替代值。请参见 “使用 ULSQLCONNECT 环境变量存储 UltraLite 参数” 一节第 46 页。</p>	X <sup>2</sup>	N/A

<sup>1</sup> 通常用户提供。

<sup>2</sup> 仅用于桌面操作系统管理工具。

<sup>3</sup> 通常硬编码或存储在文件中。

**另请参见**

- “用于 UltraLite 管理工具的连接参数的优先级” 一节第 43 页

## 在 UltraLite 连接参数中指定文件路径

设备的物理存储确定：

- 数据库是否保存为文件。
- 标识数据库时必须遵守何种命名约定。

面向单个部署平台或使用 UltraLite 桌面操作系统管理工具时，最适合使用 DBF 参数。例如：

```
ulload -c DBF=sample.udb sample.xml
```

#### Windows Mobile 提示

可以使用 UltraLite 管理工具管理已经部署到附属设备的数据库。请参见“Windows Mobile”一节第 41 页。

否则，如果编写跨平台应用程序，则使用平台特定的（CE\_FILE、NT\_FILE 或 PALM\_FILE）文件连接参数构造通用连接字符串。例如：

```
Connection = DatabaseMgr.OpenConnection("UID=JDoe;PWD=ULdb;  
CE_FILE=\database\MyCEDB.udb;PALM_FILE=MyPalmDB")
```

## Windows 桌面操作系统

桌面操作系统允许绝对或相对路径。

## Windows Mobile

Windows Mobile 设备要求所有路径是绝对路径。

可以在桌面操作系统或附属设备上管理 Windows Mobile 数据库。要在 Windows Mobile 设备上管理数据库，请确保以 **wce:\** 作为绝对路径的前缀。例如，使用 **ulunload** 实用程序：

```
ulunload -c DBF=wce:\UltraLite\myULdb.udb c:\out\ce.xml
```

在此示例中，UltraLite 将数据库从 Windows Mobile 设备卸载到 Windows 桌面操作系统文件夹 **c:\out** 的 **ce.xml** 文件中。

如果正在使用 **ulunloadold** 或 **ulunload** 实用程序直接管理 Windows Mobile 设备上的数据库，则 UltraLite 在卸载或操作发生之前无法备份数据库。必须先手工执行此操作，然后才能运行这些实用程序。

## Palm OS

Palm OS 不必使用文件路径概念。因此，如何定义它取决于存储类型（即基于记录或 VFS）。

**基于文件的存储 (VFS)** 对于 VFS 卷上的数据库，使用以下语法定义文件：

```
vfs: [ volume-label: | volume-ordinal: ] filename
```

对于内置驱动器可以将 *volume-label* 设置为 **INTERNAL**，对于扩展卡或卷的标签名可以设置为 **CARD**。对于 *volume-label* 没有缺省字符串。

或者，可以设置 *volume-ordinal* 来标识卷。由于已安装卷的枚举可能不同，请确保为所选择的内部或外部卷设置正确的卷序号。缺省值为 0（即平台枚举的第一个卷）。

对于 *filename*，总是按照 Palm OS 的文件和路径命名约定指定绝对文件路径。如果在路径中指定的目录不存在，则创建目录。

**基于记录的数据存储** 对于基于记录的数据存储，数据库名必须遵守 Palm OS 数据库名的所有约定。例如，数据库名不能超过 32 个字符限制和包含路径。

还请确保根据数据库的位置使用 **DBF** 或 **PALM\_FILE** 的适当值。

- 在设备本身以外的任何地方存储 Palm OS 数据库时（例如，使用 `uload`），对 DBF 使用扩展名 `.pdb`。
- 一旦将文件移动到设备上，HotSync 管道将删除 `.pdb` 扩展名。例如，如果在桌面操作系统上创建的数据库名为 `CustDB.pdb`，则当将它部署到设备上时，文件名更改为 `CustDB`。

### 注意

您不能使用 Palm 安装工具将数据库部署到 VFS 卷。而是，您需要将数据库直接复制到带有读卡器的介质或某些其它工具上。

### 另请参见

- [“UltraLite DBF 连接参数”一节第 212 页](#)
- [“UltraLite NT\\_FILE 连接参数”一节第 218 页](#)
- [“UltraLite CE\\_FILE 连接参数”一节第 209 页](#)
- [“UltraLite PALM\\_FILE 连接参数”一节第 221 页](#)

## 使用连接字符串打开 UltraLite 连接

连接字符串是从应用程序传递到运行库的参数集，以便可以定义和建立连接。

打开一个数据库连接之前，要执行三个步骤：

### 1. 参数定义阶段

必须通过一个所支持参数的组合定义连接。一些连接参数总是需要打开一个连接。其它连接参数用于为单个连接调整数据库功能。

提供这些参数的方式依是从 UltraLite 管理工具还是从 UltraLite 应用程序连接而变化。请参见 [“提供 UltraLite 连接参数”一节第 39 页](#)。

### 2. 字符串汇编阶段

您或应用程序将所提供的参数汇编到字符串中。连接字符串包含在分号分隔列表中定义为 `keyword=value` 对的参数列表。请参见 [“将参数装配到 UltraLite 连接字符串”一节第 43 页](#)。

例如，提供文件名、用户 ID 和口令的连接字符串片段写成以下形式：

```
DBF=myULdb.udb;UID=JDoe;PWD=token
```

### 3. 传送阶段

连接字符串汇编完后，通过 UltraLite 运行时的 UltraLite API 传递到数据库进行处理。如果连接尝试通过校验，则授予连接。如果出现以下情况，连接将失败：

- 数据库文件不存在。
- 验证不成功。

## 将参数装配到 UltraLite 连接字符串

在任何应用程序的连接代码（不论是管理工具还是自定义 UltraLite 应用程序）中提供的连接参数装配集称为**连接字符串**。在某些情况下，应用程序将 `ConnectionParms` 对象的字段分析为字符串。在其它情况下，在一行中键入一个连接字符串，并用分号分隔参数名称和值：

```
parameter1=value1;parameter2=value2
```

在使用连接字符串建立连接前，UltraLite 运行库确保将参数汇编到连接字符串中。例如，如果使用 `ulload` 实用程序，以下连接字符串用于将新的 XML 数据装载到现有数据库中。在提供此字符串之前，无法连接到指定的数据库文件：

```
ulload -c "DBF=sample.udb;UID=DBA;PWD=sql" sample.xml
```

UltraLite 在遇到无法识别的连接参数时会生成一个错误。

### 用于 UltraLite 管理工具的连接参数的优先级

所有的 UltraLite 管理工具都遵守连接参数优先级的特定顺序：

- 如果指定，`CE_FILE`、`NT_FILE` 和 `PALM_FILE` 参数总是优先于 `DBF`。
- 如果指定两个 `DBF` 参数，则最后指定的参数优先。
- 如果在连接字符串中提供重复的参数，则使用最后提供的参数。忽略所有其它参数。
- 连接字符串中的参数优先于 `ULSQLCONNECT` 环境变量或连接对象中提供的参数。
- 为在连接字符串中未提供的连接参数检查 `ULSQLCONNECT` 环境变量。
- 如果在连接字符串或 `ULSQLCONNECT` 中 `UID` 以及 `PWD` 值都未提供，则采用缺省值 `UID=DBA` 和 `PWD=sql`。

### 限制

将忽略连接字符串参数值中的所有前导空格和/或尾随空格。连接参数值都不能包含前导单引号 (')、前导双引号 (") 或分号 (;)。

### 另请参见

- “使用 `ULSQLCONNECT` 环境变量存储 UltraLite 参数” 一节第 46 页

## UltraLite 用户验证

您不能禁用 UltraLite 用户验证。一次成功的连接需要验证用户。与 SQL Anywhere 不同，创建和管理 UltraLite 数据库用户仅用于验证目的而不是用于对象所有权的目的。一旦用户通过验证并连接到数据库，该用户对该数据库中的一切内容具有不受限制的访问权限包括模式数据。

仅可以从现有连接中添加或修改 UltraLite 用户。因此，只有使用有效的用户 ID 和口令进行连接后，才能更改 UltraLite 用户库。

如果这是第一次连接，则所需的 UID 和 PWD 是与第一次创建数据库时设置的值相同的值。如果没有设置初始用户，则必须使用缺省值 **UID=DBA** 和 **PWD=sql** 进行验证。

## 跳过验证

虽然您不能禁用验证，但您可以在创建和连接数据库时使用 UltraLite 缺省值跳过验证。

如果未提供 UID 以及 PWD 参数，则 UltraLite 采用缺省值 **UID=DBA** 和 **PWD=sql**。

### ◆ 在 UltraLite 中跳过验证

1. 创建数据库时，请不要设置 UID 和 PWD 连接参数。
2. 请不要在 UltraLite 数据库中删除或修改缺省用户。
3. 连接到已创建数据库时，请不要设置 UID 和 PWD 连接参数。

### 另请参见

- [“限制”一节第 79 页](#)
- [“使用 UltraLite 用户”一节第 79 页](#)
- [“解释用户 ID 和口令组合”一节第 44 页](#)

## 解释用户 ID 和口令组合

UltraLite 允许您设置 UID 和 PWD 参数中的一个、两个或两个都不设置—当部分定义阻止用户被 UltraLite 标识的情况除外。下表介绍 UltraLite 如何解释不完整的用户定义。

如果创建数据库时使用……	它具有这样的影响……
没有用户 ID 和口令。	UltraLite 创建一个 UID 为 <b>DBA</b> 和 PWD 为 <b>sql</b> 的缺省用户。以后进行连接尝试时不需要提供这些连接参数。

如果创建数据库时使用……	它具有这样的影响……
<p>仅用户 ID 参数。</p> <p>示例：</p> <ul style="list-style-type: none"><li>● UID=JaneD</li><li>● UID=JaneD;PWD=</li><li>● UID=JaneD;PWD=""</li></ul>	<p>UltraLite 创建一个 UID 为 <b>JaneD</b> 和一个空 PWD 的缺省用户。连接时，必须始终提供 UID 参数。PWD 参数不是必需的。</p>
<p>仅口令参数。</p> <p>示例：</p> <ul style="list-style-type: none"><li>● PWD=3saBys</li><li>● UID=;PWD=3saBys</li><li>● UID="";PWD=3saBys</li></ul>	<p>UltraLite 生成一个错误。没有用户 ID，UltraLite 无法设置口令。</p>

**另请参见**

- [“UltraLite 用户验证”一节第 44 页](#)
- [“限制”一节第 79 页](#)
- [“使用 UltraLite 用户”一节第 79 页](#)

## 使用 ULSQLCONNECT 环境变量存储 UltraLite 参数

ULSQLCONNECT 环境变量是可选的，它不是由安装程序设置的。ULSQLCONNECT 包含在分号分隔的列表中定义为 *keyword=value* 对的参数列表。

使用 ULSQLCONNECT 避免在开发期间必须重复向 UltraLite 管理工具提供相同的连接参数。对于自定义应用程序，不能使用 ULSQLCONNECT。

### 小心

请不要使用英镑字符 (#) 作为等号的替代字符，在 UltraLite 中将忽略英镑字符。UltraLite 支持的所有平台允许在环境变量设置中使用 =。

### ◆ 为 UltraLite 桌面操作系统工具设置 ULSQLCONNECT

1. 运行以下命令：

```
set ULSQLCONNECT="parameter=value; ..."
```

2. 如果管理工具需要任何其它参数或您需要使用此环境变量替代缺省值设置，请确保设置这些值。用户提供的值总是优先于此环境变量。

### 另请参见

- “用于 UltraLite 管理工具的连接参数的优先级” 一节第 43 页
- “提供 UltraLite 连接参数” 一节第 39 页

### 示例

要使用 ULSQLCONNECT 连接到名为 *c:\database\myfile.udb* 的文件，并验证用户 **demo** 和口令 **test**，请在您的 ULSQLCONNECT 环境变量中设置以下变量：

```
set ULSQLCONNECT="DBF=c:\database\myfile.udb;UID=demo;PWD=test"
```

通过设置此环境变量，您不再需要为这些缺省值使用 **-c** 连接选项—除非需要替代这些值。

例如，如果使用 **ulload** 从 *extra.xml* 文件将其它信息添加到数据库，您将运行以下命令：

```
ulload -a extra.xml
```



---

# 将 UltraLite 部署到设备

## 目录

部署带 UltraLite 引擎的多个 UltraLite 应用程序 .....	49
通过 AES_FIPS 数据库加密部署 UltraLite .....	50
通过启用 TLS 的同步部署 UltraLite .....	51
部署 UltraLite HotSync 管道 .....	53
为 UltraLite 部署 ActiveSync 提供程序 .....	55
使用 ActiveSync 管理器注册应用程序 .....	57
部署 UltraLite 模式升级 .....	58

---

大多数情况下，在 Windows 桌面操作系统上进行开发，同时 UltraLite 的最终发行目标为移动设备。但是，根据部署环境，您可以使用不同的部署机制来安装 UltraLite。

UltraLite 应用程序对象可能通过同一个 UltraLite 数据库的不同迭代进行发展：开发数据库、测试数据库和已部署的生产数据库。在已部署数据库应用程序生存期内，首先在开发数据库中进行更改和提高，然后传播到测试数据库，最后分布到生产数据库。

### 初始安装

为设备初始安装上 UltraLite 解决方案是一项必需步骤，以便持续维护该设备。

### 持续维护

多数收益和开销都与正在部署且用于该领域的应用程序相关联。为此，UltraLite 支持拥有多种机制以传送应用程序和同步数据，它们两个是使 UltraLite 如此灵活的任何一个 UltraLite 部署中的主要功能。

技术	Windows Mobile	Palm OS	Java
文件传输	支持	基于记录和 VFS	
UltraLite 作为 MobiLink 同步客户端	支持	基于记录和 VFS	
关于设备文件系统浏览工具或读卡器		只有 VFS (true) <sup>1</sup>	
ActiveSync	支持	无	

技术	Windows Mobile	Palm OS	Java
HotSync	无	只有基于记录	
ALTER DATABASE SCHEMA	支持	支持	

<sup>1</sup> 如果目标是 VFS 卷，则不能使用 Palm 安装工具部署 UltraLite 数据库。而是，需要使用读卡器或某些其它工具将数据库直接复制到介质上。

**另请参见**

- [“使用 MobiLink 文件传输”一节第 129 页](#)
- [“UltraLite 客户端”第 115 页](#)
- [“Palm OS 上的 HotSync”一节第 132 页](#)
- [“Windows Mobile 上的 ActiveSync”一节第 135 页](#)
- [“部署 UltraLite 模式升级”一节第 58 页](#)
- [“UltraLite ALTER DATABASE SCHEMA FROM FILE 语句”一节第 424 页](#)

## 部署带 UltraLite 引擎的多个 UltraLite 应用程序

UltraLite 引擎是一个数据管理模块，它只在 32 位 Windows 桌面操作系统和 Windows Mobile 上管理来自应用程序的并发 UltraLite 数据库连接。该引擎由 SQL Anywhere 安装程序自动安装到桌面操作系统。因此，您只需将引擎部署到 Windows Mobile 设备。

### ◆ 将 uleng 部署到 Windows Mobile 设备

1. 复制 *uleng11.exe* 文件以及恰当的 \*.dll 文件。您复制的 \*.dll 文件应包括所有数据库加密、同步加密或所需的压缩 \*.dll 文件。

Filename	Basic	ECC TLS	RSA TLS	FIPS RSA TLS	HTT PS	Compression	FIPS AES 数据加密
<i>uleng11.exe</i>	X	X	X	X	X	X	X
<i>mlcecc11.dll</i> <sup>1</sup>		X			X		
<i>mlcrsa11.dll</i> <sup>1</sup>			X		X		
<i>mlcrsafips11.dll</i>				X	X		
<i>mlczlib11.dll</i>						X	
<i>sbgse2.dll</i>				X	X		X
<i>ulfips11.dll</i>							X
<i>ulrt11.dll</i> <sup>2</sup>	X	X	X	X	X	X	X

<sup>1</sup> 如果应用程序分别直接链接到 *ulecc.lib* 和 *ulrsa.lib*，则不需要文件。

<sup>2</sup> 如果应用程序链接到 *ulimp.lib*，则只需要文件。

2. 将文件保存到一个恰当的目录中。多数情况下，会使用以下目标目录之一：
  - `\windows` 目录。建议使用此位置，因为客户端自动在此位置中查找引擎。
  - 其它 UltraLite 应用程序文件的目录。
3. 如果您使用除了 `\windows` 目录以外的任何位置，则会包含 START 连接参数。此参数在应用程序连接到 UltraLite 数据库时启动 UltraLite 引擎。

例如，数据库的连接字符串或 Windows Mobile 客户端应用程序的连接代码可能使用此 START 参数：

```
"START=\Program Files\MyApp\uleng11.exe"
```

### 另请参见

- “通过 AES\_FIPS 数据库加密部署 UltraLite” 一节第 50 页
- “通过启用 TLS 的同步部署 UltraLite” 一节第 51 页

## 通过 AES\_FIPS 数据库加密部署 UltraLite

对数据库使用高度加密技术后，如果没有密钥（一种口令），就无法对数据库进行操作和访问。加密算法会对数据库和事务日志文件中包含的信息进行编码，以使信息无法被破译。但是，数据库加密需要使用数据库部署恰当的文件编号。

当使用 `-fips` 选项连接到 UltraLite 时，您可以运行由 AES 或 AES\_FIPS 高度加密进行加密的数据库。要确保您正在使用 AES\_FIPS 运行，使用 `-fips=1`。

如果要使用 AES FIPS 加密对数据库进行加密，必须为每个平台配置和部署设备。

### ◆ 为 AES FIPS 加密 UltraLite 数据库设置应用程序和设备

1. 使用属性 `fips=1` 创建一个 UltraLite 数据库。请参见“[UltraLite fips 创建参数](#)”一节第 176 页。
2. 在应用程序的连接字符串中使用以下连接参数：`DBKEY=key`。请参见“[UltraLite DBKEY 连接参数](#)”一节第 214 页。
3. 在 Palm OS 上，通过调用 `ULEnableRsaFipsStrongEncryption` 启用数据库加密。请参见“[ULEnableFIPSStrongEncryption 函数](#)”一节《[UltraLite - C 及 C++ 编程](#)》。
4. 在 Palm OS 上，确保除了 `ulrt.lib` 之外还链接到以下文件：
  - `ulfips.lib`
  - `gse1st.lib`
5. 确保将适当的文件部署到设备。
  - Windows 桌面操作系统和 Windows Mobile 需要 `ulfips11.dll` 和 `sbgse2.dll`。Windows Mobile 组件还需要组件 DLL 文件。
  - Palm OS 需要 `libsbgse_4i.prc`。

### 另请参见

- “[通过启用 TLS 的同步部署 UltraLite](#)”一节第 51 页
- UltraLite.NET: “[加密和模糊处理](#)”一节《[UltraLite - .NET 编程](#)》
- UltraLite for C++: “[对数据进行加密](#)”一节《[UltraLite - C 及 C++ 编程](#)》
- UltraLite for M-Business Anywhere: “[数据库加密和模糊处理](#)”一节《[UltraLite - M-Business Anywhere 编程](#)》
- “[UltraLite fips 创建参数](#)”一节第 176 页
- “[保护 UltraLite 数据库](#)”一节第 35 页

## 通过启用 TLS 的同步部署 UltraLite

### 需要单独授予许可的组成部分

ECC 加密和 FIPS 认证的加密需要单独的许可。所有高度加密技术受出口法规约束。

请参见“单独授权的组件”一节《SQL Anywhere 11 - 简介》。

MobiLink 的 UltraLite 客户端应用程序必须配置为使用启用 TLS 的同步。传送层安全支持加密、篡改检测和基于证书的验证。请参见“传送层安全简介”一节《SQL Anywhere 服务器 - 数据库管理》。

### 平台支持

RSA、ECC 和 FIPS 加密并非在所有平台上均可用。有关哪些平台支持哪种加密方法的信息，请参见 <http://www.sybase.com/detail?id=1062623>。

### ◆ 在 UltraLite 客户端应用程序和设备上设置 TLS 同步

- 通过调用您的应用程序代码中的以下函数之一启用加密同步：
  - 若要启用 RSA 加密，请调用 `UEnableRsaSyncEncryption`。请参见“`UEnableRsaSyncEncryption` 函数”一节《UltraLite - C 及 C++ 编程》。
  - 若要启用 ECC 加密，请调用 `UEnableEccSyncEncryption`。请参见“`UEnableEccSyncEncryption` 函数”一节《UltraLite - C 及 C++ 编程》。
  - 若要启用 FIPS RSA 加密，请调用 `UEnableRsaFipsSyncEncryption`。只有 Palm OS 客户端才需要此函数。请参见“`UEnableRsaFipsSyncEncryption` 函数”一节《UltraLite - C 及 C++ 编程》。
- 将同步信息流设置为 TLS 或 HTTPS。
- 如果要启用 ECC 或 FIPS 加密，还必须：
  - **ECC** 将 `tls_type` 网络协议选项设置到 **ECC**。请参见“`tls_type`”一节《MobiLink - 客户端管理》。
  - **FIPS** 将 `fips` 网络协议选项设置到 **Yes**。请参见“`fips`”一节《MobiLink - 客户端管理》。
- 确保您已链接到适当的库：

平台	链接	RSA 加密	ECC 加密	FIPS 加密
Windows 桌面操作系统	静态 <sup>1</sup>	<i>ulrsa.lib</i>	<i>ulecc.lib</i>	无
Windows 桌面操作系统	动态 <sup>2</sup>	无	无	无
Windows Mobile	静态 <sup>1</sup>	<i>ulrsa.lib</i>	<i>ulecc.lib</i>	无

平台	链接	RSA 加密	ECC 加密	FIPS 加密
Windows Mobile	动态 <sup>2</sup>	无	无	无
Palm OS	静态 <sup>1</sup>	<i>ulrsa.lib</i>	<i>ulecc.lib</i>	<i>ulfips.lib</i> 、 <i>gse1st.lib</i>

<sup>1</sup> 您还必须链接到 *ulrt.lib*。

<sup>2</sup> 您还必须链接到 *ulimp.lib*。

5. 确保将适当的文件复制到设备：

平台	链接	RSA 加密	ECC 加密	FIPS 加密
Windows 桌面操作系统	静态	无	无	<i>mlcrsafips11.dll</i> <i>sbgse2.dll</i>
Windows 桌面操作系统	动态 <sup>1</sup>	<i>mlcrsa11.dll</i>	<i>mlcecc11.dll</i>	<i>mlcrsafips11.dll</i> <i>sbgse2.dll</i>
Windows Mobile	静态	无	无	<i>mlcrsafips11.dll</i> <i>sbgse2.dll</i>
Windows Mobile	动态 <sup>1</sup>	<i>mlcrsa11.dll</i>	<i>mlcecc11.dll</i>	<i>mlcrsafips11.dll</i> <i>sbgse2.dll</i>
Palm OS	静态	无	无	<i>libsbgse_4i.prc</i>
Windows Mobile 组件和 UltraLite 引擎	静态 <sup>2</sup>	<i>mlcrsa11.dll</i>	<i>mlcecc11.dll</i>	<i>mlcrsafips11.dll</i> <i>sbgse2.dll</i>

<sup>1</sup> 您还必须部署 *ulrt11.dll*。

<sup>2</sup> 您还必须部署您的组件 *.dll* 文件和/或 *uleng11.exe*。

### 另请参见

- “配置 UltraLite 客户端以使用传送层安全”一节 《SQL Anywhere 服务器 - 数据库管理》
- UltraLite.NET: “加密和模糊处理”一节 《UltraLite - .NET 编程》
- UltraLite for C++: “对数据进行加密”一节 《UltraLite - C 及 C++ 编程》
- UltraLite for M-Business Anywhere: “数据库加密和模糊处理”一节 《UltraLite - M-Business Anywhere 编程》

## 部署 UltraLite HotSync 管道

UltraLite HotSync 管道是一个软件模块，它使用户可以从桌面访问其设备。像其它软件组件一样，您需要部署设备的必要文件以确保 UltraLite 通过 Windows Mobile ActiveSync 或 Palm OS HotSync 软件运行。

在开发期间，您使用 SQL Anywhere 安装程序将 UltraLite 安装到桌面操作系统上。但是，然后还需要在最终用户的计算机上部署必需的 HotSync 管道文件。必须在计算机上安装 HotSync 管理器以便安装程序检测软件并运行。

### UltraLite HotSync 管道文件

- ***install-dir\Bin32\Condmgr\condmgr.dll*** 实用程序 DLL，用于查找 HotSync 安装路径及通过 HotSync 注册管道。
- ***install-dir\Bin32\ulcond11.exe*** UltraLite HotSync 管道安装实用程序，用于在台式计算机上安装或删除 UltraLite HotSync 管道。请参见“[Palm OS 的 UltraLite HotSync 管道安装实用程序 \(ulcond11\)](#)”一节第 243 页。
- ***install-dir\Bin32\dbhsync11.dll*** 由 HotSync 调用的 DLL 管道。
- ***install-dir\Bin32\dblgen11.dll*** 语言资源库。对于英语之外的语言，将此文件名中的字母 *en* 替换为该语言的两个字母的缩写，例如：对德语使用 *dblgede11.dll*，而对日语则使用 *dblgeja11.dll*。
- **流 DLL** 可选。在 UltraLite HotSync 管道和 MobiLink 服务器之间进行的加密网络通信需要流 DLL。
  - 对于使用 TLS 和 HTTPS 加密 RSA，为 *install-dir\Bin32\mlcrsa11.dll*。
  - 对于使用 TLS 和 HTTPS 加密 ECC，为 *install-dir\Bin32\mlcecc11.dll*。
  - 对于使用 TLS 和 HTTPS 加密 RSA FIPS，为 *install-dir\Bin32\mlcrsafips11.dll*。

#### 需要单独授予许可的组成部分

ECC 加密和 FIPS 认可的加密流 DLL 需要单独的许可。所有高度加密技术受出口法规约束。

要订购单独授权的组件，请访问 [http://www.iAnywhere.com/products/separately\\_licensed\\_components.html](http://www.iAnywhere.com/products/separately_licensed_components.html)。

有关组件和平台支持的信息，请参见 <http://www.sybase.com/detail?id=1062617>。

### ◆ 部署和注册 UltraLite HotSync 管道

1. 在最终用户的桌面操作系统上，创建以下目录：
  - *MyDir\win32*
  - *MyDir\win32\condmgr*
2. 将以下文件的副本部署到 *MyDir\win32* 目录中：
  - *ulcond11.exe*
  - *dbhsync11.dll*
  - *dblgen11.dll*

3. 将文件 *Condmgr.dll* 的副本部署到 *MyDir\win32\condmgr* 目录中。

4. 查找以下注册表项：

```
HKEY_CURRENT_USER\Software\Sybase\SQL Anywhere\11.0\
```

5. 在此注册表项中创建一个名为 **Location** 的值，并将该值数据设置为管道的根部署文件夹。例如 *MyDir*。

6. 如果最终用户需要证书来加密通信流，则在台式计算机上安装根证书以便管道能够访问它。

7. 运行 **ulcond11** 以确保已使用 **-c** 选项（可能还有 **-a** 选项）为每个 UltraLite 数据库设置了连接字符串。还必须设置正确的创建者 ID。

此实用程序部署并正确配置了 UltraLite HotSync 管道。

### 提示

如果使用的是加密密钥，请避免在连接字符串中设置密钥。此位置可能带来安全风险。相反，允许管道提示用户输入密钥。

例如，以下命令为创建者 ID 为 Syb2 的应用程序安装名为 CustDB 的管道。

```
ulcond11 -c "DBF=custdb.udb;UID=DBA;PWD=sql" -n CustDB Syb2
```

8. 如果未在 UltraLite 应用程序的 *ul\_synch\_info* 结构中包括同步参数，则在 HotSync 中或使用 *ulcond11* 配置此信息。请参见“[设置 MobiLink 同步的协议选项](#)”一节第 133 页。

### ◆ 检查是否正确部署了 HotSync 管道

● 在计算机的系统任务栏中，右击 [**HotSync 管理器**] 并选择 [**自定义**]。

为每个 HotSync 用户显示管道列表。验证在继续启动 MobiLink 服务器并测试同步操作之前已列出管道。



## 为 UltraLite 部署 ActiveSync 提供程序

UltraLite ActiveSync 提供程序是一个软件模块，它使用户可以从桌面访问其设备。像其它软件组件一样，您需要部署设备的必要文件以确保 UltraLite 通过 Windows Mobile ActiveSync 或 Palm OS HotSync 软件运行。

在开发期间，您使用 SQL Anywhere 安装程序将 UltraLite 安装到桌面操作系统上。但是，将 UltraLite 部署到最终用户时，必须在最终用户的计算机上手工安装和注册 ActiveSync 提供程序。此要求可确保 ActiveSync 知道何时为特定应用程序调用某个提供程序的特定实例。

- **mlasinst.exe** 安装 ActiveSync 提供程序并使用 ActiveSync 管理器注册。该实用程序也使用 ActiveSync 提供程序注册应用程序以进行同步。
- **mlasdesk.dll** 该 DLL 由 ActiveSync 管理器装载到桌面操作系统上。*mlasinst.exe* 使用 ActiveSync 管理器注册此文件的位置。
- **mlasdev.dll** 该 DLL 由 ActiveSync 管理器装载到设备上。*mlasinst.exe* 将此文件部署到该设备上的正确位置。
- **dbigen11.dll** 语言资源库。

有关支持的提供程序平台的列表，请参见 <http://www.sybase.com/detail?id=1062617>。

### ◆ 安装 ActiveSync 应用程序

1. 确保最终用户：
  - 已安装 ActiveSync 管理器。
  - 已将 ActiveSync 提供程序文件从某台开发计算机复制到该用户的硬盘。
2. 运行 *mlasinst* 为 ActiveSync 安装一个提供程序。也可以使用它进行注册，以及将 UltraLite 应用程序部署到用户的 Windows Mobile 设备上—取决于所使用的命令行语法。如果 UltraLite 应用程序使用多个文件，则必须手工复制所需的文件。

以下示例假定 *mlasdesk.dll* 和 *mlasdev.dll* 都位于当前目录中。使用 *-k* 和 *-v* 选项。*-p* 和 *-x* 选项是应用程序由 ActiveSync 启动时的命令行选项。

```
mlasinst "C:\My Files\myULapp.exe" "\Program Files\myULapp.exe"
        "My Application" MYAPP -p -x -v -k
```

如果您使用此实用程序为 ARM 5.0 处理器部署预编译的 CustDB，则命令行应类似于以下内容：

```
mlasinst -v "install-dir\UltraLite\ce\arm.50"
        "install-dir\UltraLite\ce\arm.50\custdb.exe" custdb.exe CustDB CUSTDBDEMO
```

#### 注意

也可在以后使用 ActiveSync 注册 UltraLite 应用程序（如果您选择这样做）。请参见“使用 ActiveSync 管理器注册应用程序”一节第 57 页。

3. 重新启动计算机，以便 ActiveSync 能够识别新的提供程序。
4. 启用 MobiLink 提供程序。

- a. 从 ActiveSync 窗口，单击 [选项]。
- b. 检查列表中的 **MobiLink 客户端**，并单击 [确定] 以激活提供程序。
- c. 要查看已注册应用程序的列表，请单击 [选项]，选择 **MobiLink 客户端**，并单击 [设置]。

**另请参见**

- “使用 ActiveSync 管理器注册应用程序” 一节第 57 页
- “ActiveSync 提供程序的安装实用程序 (mlasinst)” 一节 《MobiLink - 客户端管理》

## 使用 ActiveSync 管理器注册应用程序

注册与 ActiveSync 一起使用的应用程序的方法有以下两种：使用 ActiveSync 提供程序安装实用程序，或使用 ActiveSync 管理器本身。本节介绍如何使用 ActiveSync 管理器。

### ◆ 注册与 ActiveSync 一起使用的应用程序

1. 启动 ActiveSync。
2. 从 ActiveSync 窗口，选择 [选项]。
3. 从信息类型列表中，选择 **MobiLink 客户端** 并单击 [设置]。
4. 在 **MobiLink 同步** 窗口中，单击 [新建]。
5. 为应用程序输入以下信息：
  - **应用程序名** 标识显示于 ActiveSync 用户界面中的应用程序的名称。
  - **类名** 应用程序的注册类名。请参见“为应用程序指派类名”一节《UltraLite - C 及 C++ 编程》。
  - **路径** 应用程序在设备上的位置。
  - **参数** ActiveSync 启动应用程序时使用的任何命令行参数。
6. 单击 [确定] 注册该应用程序。

### 另请参见

- “ActiveSync 提供程序的安装实用程序 (mlasinst)”一节《MobiLink - 客户端管理》

## 部署 UltraLite 模式升级

要进行模式升级，使用 SQL 语句 ALTER DATABASE SCHEMA FROM FILE。

### 升级过程

#### 小心

在模式升级过程中不要重置设备。如果在模式升级过程中重置了设备，则数据将会丢失且 UltraLite 数据库标记为 "bad"。

1. 比较新数据库模式和现有数据库模式的不同之处。
2. 现有数据库的模式会发生变更。
3. 删除不适合新模式的行。例如：
  - 如果将唯一性约束添加到表并有多行拥有相同的值，则将删除除一行之外的所有行。
  - 如果您尝试更改列域并出现转换错误时，则将删除那一行。例如，如果您有一列 VARCHAR 列并将其转换为 INT 列，而某一行的值是 ABCD，则删除该行。
  - 如果新模式拥有新的外键，其中外行没有匹配的主键，则删除这些行。
4. 删除行时会出现 SQLE\_ROW\_DROPPED\_DURING\_SCHEMA\_UPGRADE (130) 警告。

#### ◆ 升级 UltraLite 模式

1. 创建一个定义新模式的 DDL 语句的 SQL 脚本。SQL 脚本文件的字符集必须匹配您想升级的数据库的字符集。

应使用 `ulinit` 或 `ulunload` 以抽取您的脚本所需的 DDL 语句。通过使用这些带以下选项的实用程序，确保 DDL 语句在语法上是正确的。

- 如果您正在使用 `ulunload`，则使用 `-n` 和 `-s [file]` 选项。
- 如果您正在使用 `ulinit`，则使用 `-l [file]` 选项。

如果您未使用 `ulunload` 和 `ulinit`，则查看脚本以确保：

- 未重命名表、列或发布。不支持 RENAME 操作。如果重命名表，则作为 DROP TABLE 和 CREATE TABLE 操作处理。
  - 未包含非 DDL 语句。包含非 DDL 语句可能不具有您预期的效果。
  - SQL 语句中的字以空格分隔。
  - 一行只能显示一条 SQL 语句。
  - 用双连字符 (-) 作为注释的前缀，并只出现在行首。
2. 针对哪个数据库将执行升级，备份该数据库。
  3. 运行新语句。例如：

```
ALTER DATABASE SCHEMA FROM FILE 'MySchema.sql';
```

## 错误通知

因为在升级过程中激活了 Ultralite 错误回调，所以在转换过程中通知您有错误。例如，SQLE\_CONVERSION\_ERROR 报告不能以其参数进行转换的所有值。错误不表示过程失败。在这种情况下，返回语句后的最终 SQL 代码是 130 警告。这些警告描述了转换过程的操作，没有停止升级过程。

## 另请参见

- [“UltraLite ALTER DATABASE SCHEMA FROM FILE 语句”一节第 424 页](#)
- [“UltraLite 初始化数据库实用程序 \(ulinit\)”一节第 249 页](#)
- [“UltraLite 卸载旧数据库实用程序 \(ulunload\)”一节第 260 页](#)
- [“UltraLite SQL 语句”第 421 页](#)
- [“UltraLite 中的注释”一节第 283 页](#)

---

---

# 使用 UltraLite 数据库

## 目录

使用 UltraLite 表和列 .....	62
使用 UltraLite 索引 .....	71
使用 UltraLite 发布 .....	75
使用 UltraLite 用户 .....	79
使用事件通知 .....	81
SQL 直通的 UltraLite 支持 .....	83

---

## 使用 UltraLite 表和列

可使用表来存储数据并定义表中数据的关系。表由行和列组成。每一列保存一类特定的信息（例如电话号码或名称），而每一行指定一个特定的条目。

首次创建 UltraLite 数据库时，您只能看到系统表。系统表中保存了 UltraLite 模式。您可以根据需要在 Sybase Central 中隐藏或显示这些表。

您还可以根据应用程序的需要添加新表。您也可以浏览这些表中的数据，在源数据库的现有表或是其它打开的目标数据库之间复制和粘贴数据。

### 行压缩和表定义

UltraLite 使用两种格式的行：

- **解压缩行** 为解压缩格式。必须先解压缩每一行，之后才可读取或写入各个列值。
- **压缩行** 为解压缩行的压缩形式，其中每一个列值都被压缩了，以使整个行占用的内存尽可能小。压缩行的大小完全取决于每一列中的值：例如，两行可属于同一表，但它们的压缩大小却有很大不同。注意，LONG BINARY 和 LONG VARCHAR 列是与压缩行分开存储的。

UltraLite 有一个限制，那就是压缩行必须适合数据库页面的大小。由于 LONG BINARY 和 LONG VARCHAR 列不与压缩行一起存储，因此，它们可超出页面大小。

表定义在 UltraLite 运行时压缩数据之前描述行，理解这一点很重要。由于压缩行的大小取决于每一列中的值，因此不管压缩行的要求是否得到了满足，您都无法通过表定义很容易地进行预先确定。鉴于此，UltraLite 允许您定义一个表，该表中的解压缩行将不会适合页面。欲了解某一行是否适合页面，您必须尝试着插入或更新该行本身，如果行不适合页面，UltraLite 会检测并报告此错误。

#### 注意

不能将表声明为您所需的任何大小。UltraLite 所允许的已声明表行大小限制为 64 KB。如果您试图定义一个解压缩行超出此上限的表，UltraLite 则会生成一个 SQL 错误代码 `SQL_MAX_ROW_SIZE_EXCEEDED (-1132)`。

### 另请参见

- “UltraLite page\_size 创建参数” 一节第 182 页
- “数据库表” 一节 《SQL Anywhere 11 - 简介》
- “在 SQL Anywhere 中创建数据库” 《SQL Anywhere 服务器 - SQL 的用法》
- “UltraLite 系统表” 第 267 页

## 创建 UltraLite 表

可以在 Interactive SQL 中使用 SQL 语句或通过 Sybase Central 来创建用于保存关系数据的新表。

在 UltraLite 中，您只能创建基表，您将对表进行声明以保存持久数据。在您显式删除数据或删除该表之前，基表及其数据都一直存在。UltraLite 不支持全局临时表或已声明临时表。



**注意**

UltraLite 应用程序中的表必须包含一个主键。在 MobiLink 同步过程中也需要主键，以便将 UltraLite 数据库中的行与统一数据库中的行关联。

## Sybase Central

在 Sybase Central 中，您可在使用所选数据库时执行这些任务。

### ◆ 创建 UltraLite 表 (Sybase Central)

1. 连接到 UltraLite 数据库。
2. 在左窗格中，右击 [表] 并选择 [新建] » [表]。
3. 在 [您要给新表指定什么名称] 字段中键入新表名称。
4. 单击 [完成]。
5. 从 [文件] 菜单中选择 [保存表]。

## Interactive SQL

在 Interactive SQL 中，您可以在创建新表时声明列。

### ◆ 创建 UltraLite 表 (Interactive SQL)

1. 连接到 UltraLite 数据库。
2. 执行 CREATE TABLE 语句。

例如，以下语句会创建一个新表来描述公司中雇员的各种技能和专业水平。表中的列分别保存每种技能的标识号、技能名称和类型（例如，技术或管理）。

```
CREATE TABLE Skills (  
    SkillID INTEGER PRIMARY KEY,  
    SkillName CHAR( 20 ) NOT NULL,  
    SkillType CHAR( 20 ) NOT NULL  
);
```

## 另请参见

- [“UltraLite CREATE TABLE 语句”一节第 440 页](#)
- [“在 UltraLite 表中添加列”一节第 64 页](#)

## 使用 allsync 和 nosync 后缀

您可以将 `_allsync` 或 `_nosync` 追加到表名称的后面以便控制同步的数据限制。您可以使用这些后缀作为使用发布来控制数据限制一种替代方法。要控制数据优先级，需定义一个或多个发布。

- 如果所创建表的名称以 **\_allsync** 结尾，即使自上次同步以来未更改该表中的所有行，每次同步时也将同步这些行。

**提示**

您可将用户特定的数据或客户端特定的数据存储在 **allsync** 表中。然后可以在同步时将 UltraLite 表中的数据上载到统一数据库的临时表中。同步脚本可以控制数据并使您不必在统一数据库中维护该数据。

- 如果所创建表的名称以 **\_nosync** 结尾，则该表的所有行均不会同步。可使用这些表来保存统一数据库的表中不需要的持久数据。

**另请参见**

- “使用 UltraLite 发布” 一节第 75 页
- “设计 UltraLite 中的同步” 一节第 123 页
- “UltraLite 中的 nosync 表” 一节第 124 页
- “UltraLite 中的 allsync 表” 一节第 124 页
- “UltraLite CustDB 示例” 第 85 页

**示例**

在 *CustDB.udb* 示例数据库中，您会发现，已将一个表声明为 **nosync** 表，因为该表名已命名为 **ULIdentifyEmployee\_nosync**。因此，无论该表中的数据如何更改，它都不会与 **MobiLink** 同步，并且信息也不会出现在 *CustDB.db* 统一数据库中。

## 在 UltraLite 表中添加列

如果表是空的，您可以很容易地添加新列。但是，如果表中已存在数据，且列定义包含缺省值或允许 **NULL** 值，则您只能添加一列。

您可以使用 **Sybase Central** 来执行此任务，也可以通过执行一个 **SQL** 语句（例如，**Interactive SQL**）来执行此任务。

### **Sybase Central**

在 **Sybase Central** 中，您可在使用所选表时执行此任务。

◆ **向 UltraLite 表中添加新列 (Sybase Central)**

1. 连接到 UltraLite 数据库。
2. 在左窗格中，双击 [表]。
3. 双击表。
4. 单击 [列] 选项卡，右击表下的空格并选择 [新建] » [列]。
5. 设置新列的属性。
6. 从 [文件] 菜单中选择 [保存表]。

## Interactive SQL

在 Interactive SQL 中，您只能在创建表或更改表时声明列。

### ◆ 向新 UltraLite 表中添加列 (Interactive SQL)

1. 连接到 UltraLite 数据库。
2. 执行 CREATE TABLE 语句或 ALTER TABLE 语句，并通过声明名称和其它属性来定义列。

以下示例为图书馆数据库创建用于保存借出图书信息的表。date\_borrowed 的缺省值指示在创建条目的当天借出图书。date\_returned 列在归还图书前一直为 NULL。

```
CREATE TABLE borrowed book (
  loaner_name CHAR(100) PRIMARY KEY,
  date_borrowed DATE NOT NULL DEFAULT CURRENT DATE,
  date_returned DATE,
  book CHAR(20)
);
```

以下示例将 customer 表修改为包含一个最多可容纳 50 个字符的地址列：

```
ALTER TABLE customer
ADD address CHAR(50);
```

### 另请参见

- “选择对象名称”一节 《SQL Anywhere 服务器 - SQL 的用法》
- “UltraLite 中的数据类型”一节第 290 页
- “选择列数据类型”一节 《SQL Anywhere 服务器 - SQL 的用法》
- “UltraLite CREATE TABLE 语句”一节第 440 页
- “UltraLite ALTER TABLE 语句”一节第 428 页

## 变更 UltraLite 列定义

您可以通过修改各种列属性（甚至于彻底删除列）来更改表的列定义结构。修改后的列定义必须满足列中所存储的所有数据的要求。例如，如果列中已存在一个 NULL 条目，则您不能将该列修改为禁止 NULL。

可以使用 Sybase Central 或 Interactive SQL 执行此任务。

### Sybase Central

在 Sybase Central 中，您可在使用所选表时执行这些任务。

#### ◆ 变更现有的 UltraLite 列 (Sybase Central)

1. 连接到 UltraLite 数据库。
2. 在左窗格中，双击 [表]。
3. 双击表。
4. 单击 [列] 选项卡并更改列属性。

5. 从 [文件] 菜单中选择 [保存表]。

## Interactive SQL

在 Interactive SQL 中，您可以使用 ALTER TABLE 语句执行这些任务。

### ◆ 变更现有的 UltraLite 列 (Interactive SQL)

1. 连接到 UltraLite 数据库。
2. 执行 ALTER TABLE 语句。

以下这些示例说明如何更改数据库的结构。在所有的这些情况下，都会立即提交语句。因此，一旦进行了更改，引用该表的所有项都可能无法工作。

以下语句将 SkillDescription 列从最多 254 个字符缩短为最多 80 个字符：

```
ALTER TABLE Skills
MODIFY SkillDescription CHAR( 80 );
```

以下语句删除 Classification 列：

```
ALTER TABLE Skills
DROP Classification;
```

以下语句更改整个表的名称：

```
ALTER TABLE Skills
RENAME Qualification;
```

## 另请参见

- “选择对象名称”一节 《SQL Anywhere 服务器 - SQL 的用法》
- “UltraLite 中的数据类型”一节第 290 页
- “选择列数据类型”一节 《SQL Anywhere 服务器 - SQL 的用法》
- “UltraLite ALTER TABLE 语句”一节第 428 页

## 删除 UltraLite 表

如果表满足以下条件，即可将其删除：

- 未用作发布中的项目。
- 不存在任何被其它表的外键所引用的列。

在这些情况下，必须先更改发布或删除外键，之后才可成功删除表。

可以使用 Sybase Central 或 Interactive SQL 执行此任务。

## Sybase Central

在 Sybase Central 中，您可在使用所选表时执行这些任务。

#### ◆ 删除 UltraLite 表 (Sybase Central)

1. 连接到 UltraLite 数据库。
2. 在左窗格中，双击 [表]。
3. 右击表，然后选择 [删除]。
4. 单击 [是]。

### Interactive SQL

在 Interactive SQL 中，删除 (Delete) 表也称作将其删除 (Drop)。您可以通过执行 DROP TABLE 语句来删除表。

#### ◆ 删除 UltraLite 表 (Interactive SQL)

1. 连接到 UltraLite 数据库。
2. 执行 DROP TABLE 语句。

例如，以下 DROP TABLE 语句会删除 Skills 表中的所有记录，然后从数据库中删除 Skills 表的定义：

```
DROP TABLE Skills;
```

与 CREATE 语句类似，DROP 语句在删除表前后会自动执行 COMMIT 语句。此语句会使上次执行 COMMIT 或 ROLLBACK 后对数据库的所有更改成为永久更改。DROP 语句还会删除表上的所有索引。

### 另请参见

- [“UltraLite DROP TABLE 语句”一节第 448 页](#)

## 浏览 UltraLite 表中的信息

您可以使用 Sybase Central 或 Interactive SQL 来浏览保存在 UltraLite 数据库表中的数据。表可以是用户表或系统表。您可以通过从数据库的当前视图中显示和隐藏系统表来过滤表。由于 UltraLite 没有所有者这一概念，因此，所有用户均可浏览全部表。

### Sybase Central

在 Sybase Central 中，您可在使用所选数据库时执行这些任务。

#### ◆ 浏览 UltraLite 表 (Sybase Central)

1. 连接到 UltraLite 数据库。
2. 如果系统表被隐藏了，而你想要浏览一个或多个表中的数据，请右击 [内容] 窗格中的空格并选择 [显示系统对象]。
3. 要查看表的列表，请双击 [表]。

4. 要查看表数据，请双击表并单击右侧窗格中的 [数据] 选项卡。

◆ **过滤 UltraLite 系统表 (Sybase Central)**

1. 连接到 UltraLite 数据库。
2. 右击您所连接到的数据库并选择 [隐藏系统对象] 或 [显示系统对象]。

**Interactive SQL**

在 Interactive SQL 中，您可以使用 SELECT 语句来执行这些任务。

◆ **浏览 UltraLite 用户表 (Interactive SQL)**

1. 连接到数据库。
2. 执行 SELECT 语句，指定您要浏览的用户表。

◆ **浏览 UltraLite 系统表 (Interactive SQL)**

1. 连接到数据库。
2. 通过您要浏览的系统表来执行 SELECT 语句。

例如，要在 Interactive SQL 中显示 [结果] 窗格 [结果] 选项卡中表 systable 的内容，请执行以下命令：

```
SELECT * FROM SYSTABLE;
```

**另请参见**

- [“UltraLite 系统表” 第 267 页](#)

## 将数据复制和粘贴到 UltraLite 数据库中或从中复制和粘贴数据

在 Sybase Central 中，可以执行复制和粘贴以及拖放操作。这种数据转移使您可以在一个或多个数据库之间共享或移动对象。通过复制和粘贴操作或拖放操作，您可以共享下面的表所描述的数据。

目标	结果
另外的 UltraLite 或 SQL Anywhere 数据库。	这样，一个新对象就会被创建，并且原始对象的代码将被复制到新对象中。
同一个 UltraLite 数据库。	创建完对象的副本后，您必须重命名新对象。

**注意**  
您可以从于 MobiLink 中打开的数据库复制数据并将其粘贴到 UltraLite 数据库中。然而，您不能将 UltraLite 数据粘贴到于 MobiLink 中打开的数据库中。

## Sybase Central

在 UltraLite 插件中复制以下任何对象时，该对象的 SQL 也将被复制到剪贴板上。之后可将此 SQL 粘贴到其它应用程序中，如 Interactive SQL 或文本编辑器。例如，如果在 Sybase Central 中复制某个索引，然后将其粘贴到文本编辑器中，则会出现与该索引相应的 CREATE INDEX 语句。可以在 UltraLite 插件中复制以下对象：

- 项目
- 列
- 外键
- 索引
- 发布
- 表
- 唯一约束

## Interactive SQL

使用 Interactive SQL，您还可以将数据从结果集中复制到另一个对象中。

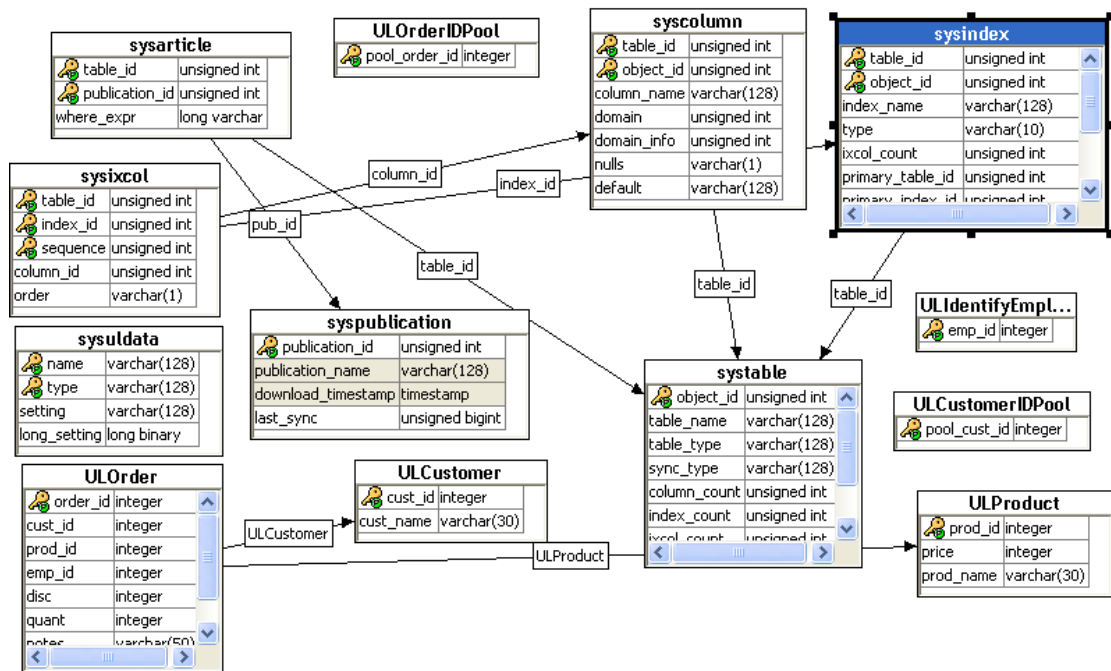
- 在指定的对象中使用 SELECT 语句结果。
- 使用 INSERT 语句将一行或选择的多行从数据库的其它位置插入表中。

## 另请参见

- [“在 SQL Anywhere 插件中复制数据库对象”一节 《SQL Anywhere 服务器 - 数据库管理》](#)
- [“UltraLite INSERT 语句”一节第 451 页](#)
- [“UltraLite SELECT 语句”一节第 457 页](#)

## 从 UltraLite 插件查看实体关系图

从 UltraLite 插件连接到数据库后，可以查看数据库中各表的实体关系图。选定数据库后，单击右窗格中的 **[ER 图]** 选项卡即可查看实体关系图。



重新排列图中的对象时，所做更改在各 Sybase Central 会话间会一直保持。双击某个表会得到该表的列定义。

### 另请参见

- “在 SQL Anywhere 中创建数据库” 《SQL Anywhere 服务器 - SQL 的用法》



## 使用 UltraLite 索引

索引根据一个或多个列中的值对表的行进行排序（升序或降序）。当 UltraLite 优化查询时，其会通过扫描现有的索引来查看是否存在查询中指定的表。如果有助于 UltraLite 更快地返回行，将使用索引。如果您在应用程序中使用 UltraLite Table API，那么，您可以指定一个索引，该索引可帮助确定行的遍历顺序。

### 性能提示

索引可以提高查询的性能—尤其对于大表来说，更是如此。要查看查询是否使用了特定的索引，您可以使用 Interactive SQL 来检查执行计划。

或者，您的 UltraLite 应用程序可以包含具有返回计划方法的 PreparedStatement 对象。

### 关于复合索引

多列索引有时也称为复合索引。利用索引中的附加列，您可以缩小搜索的范围，但使用一个具有两列的索引不同于使用两个单独的索引。例如，以下语句会创建一个具有两列的复合索引：

```
CREATE INDEX name
ON Employees ( Surname, GivenName );
```

如果第一列不能单独提供较高的选择性，复合索引将会非常有用。例如，如果许多雇员都同姓，则 Surname 和 GivenName 上的复合索引非常有用。因为每个雇员都有唯一的 ID，所以 EmployeeID 和 Surname 上的复合索引可能没有用处，因此列 Surname 不会提供任何附加选择性。

### 另请参见

- “使用索引扫描”一节第 100 页
- “UltraLite 中的执行计划”一节第 320 页
- “复合索引”一节 《SQL Anywhere 服务器 - SQL 的用法》
- UltraLite.NET: “使用 Table API 访问和操作数据”一节 《UltraLite - .NET 编程》
- UltraLite.NET: “Prepare 方法”一节 《UltraLite - .NET 编程》
- UltraLite for C++: “使用 Table API 访问数据”一节 《UltraLite - C 及 C++ 编程》
- UltraLite for C++: “UltraLite\_PreparedStatement 类”一节 《UltraLite - C 及 C++ 编程》
- UltraLite for M-Business Anywhere: “使用 Table API 处理数据”一节 《UltraLite - M-Business Anywhere 编程》
- UltraLite for M-Business Anywhere: “PreparedStatement 类”一节 《UltraLite - M-Business Anywhere 编程》

## 何时使用索引

在以下情况下使用索引：

- **您想要 UltraLite 维护参照完整性** 索引还使 UltraLite 能够对表中的行强制实施唯一性约束。不必为非常类似的数据添加索引。

- **特定查询的性能对您的应用程序来说是很重要的** 如果某个索引提高了查询的性能，并且查询的性能对您的应用程序来说很重要且经常使用，那么，您就希望维护该索引。除非所涉及的表非常小，否则，索引都可以显著提高搜索性能。因此通常建议您在频繁执行数据搜索时使用索引。
- **存在复杂查询** 对于更为复杂的查询（例如，那些具有 JOIN、GROUP BY 和 ORDER BY 子句的查询），使用索引会获得更大的改进—尽管很难确定性能的改进程度如何。因此，请分别在使用和不使用索引这两种情况下测试您的查询，以确定哪个性能更好。
- **UltraLite 表很大** 查找行所用的平均时间会随表大小的增加而增加。因此，如果要在一个非常大的表中增强搜索性能，请考虑使用索引。索引可以让 UltraLite 迅速地查找行—但这仅限于被索引的列。否则，UltraLite 必须搜索表中的每一行以查看该行是否与搜索条件相匹配，处理大表时这可能会很费时。
- **UltraLite 客户端应用程序不执行大量的插入、更新或删除操作** 由于 UltraLite 维护索引和数据本身，因此，此上下文中的索引会对数据库操作的性能产生负面影响。鉴于此，您应当将索引的使用限定于定期查询的数据（如上面一条中所述）。维护 UltraLite 缺省索引（主键和唯一约束的索引）就足够了。
- **使用 WHERE 子句和/或 ORDER BY 子句中涉及的列索引。** 这些索引可加快对这些子句的评估。特别是，索引有助于优化多列 ORDER BY 子句—但只有当索引中列的放置和 ORDER BY 子句的放置完全相同时才会如此。

## 选择索引类型

UltraLite 支持不同类型的索引：唯一键、唯一索引和非唯一索引。各个索引之间的区别在于索引所允许的内容不同。

索引特性	唯一键	唯一索引	非唯一索引
允许索引列中具有相同值的行具有重复的索引条目。	否	否	是
允许索引列中存在空值。	否	支持	支持

**注意**

您可以创建唯一键的外键，而不是唯一索引的外键。

同样，没有必要在键列上手工创建索引，通常也不建议这样做。UltraLite 为唯一键自动创建和维护索引。

**另请参见**

- “添加 UltraLite 索引” 一节第 72 页

## 添加 UltraLite 索引

可以使用 Sybase Central 或 Interactive SQL 执行此任务。

**注意**

UltraLite 不检测重复索引或多余的索引。由于索引必须与数据库中的数据一起进行维护，因此，添加索引时需谨慎。

**Sybase Central**

在 Sybase Central 中，您可在使用所选数据库时执行此任务。

**◆ 为给定的 UltraLite 表创建新索引 (Sybase Central)**

1. 连接到 UltraLite 数据库。
2. 右击 [索引]，然后选择 [新建] » [索引]。
3. 请按照向导中的说明进行操作。

**Interactive SQL**

在 Interactive SQL 中，您可以使用 CREATE INDEX 语句执行该任务。

**◆ 为给定的 UltraLite 表创建新索引 (Interactive SQL)**

1. 连接到 UltraLite 数据库。
2. 执行 CREATE INDEX 语句。

此语句会使用您配置的缺省最大散列大小来创建索引。为了能够创建一个可替换缺省值的索引，请确保使用 WITH MAX HASH SIZE *value* 子句以便为该索引实例设置一个新值。请参见“[UltraLite CREATE INDEX 语句](#)”一节第 433 页。

例如，为加快对数据库（用于跟踪雇员信息）中雇员的姓的搜索速度，以及为了针对此索引调整查询性能，您可以使用以下语句创建一个名为 EmployeeNames 的索引并将散列大小增加至 20 个字节：

```
CREATE INDEX EmployeeNames
ON Employees (Surname, GivenName)
WITH MAX HASH SIZE 20;
```

**另请参见**

- “[UltraLite CREATE INDEX 语句](#)”一节第 433 页

## 删除索引

删除索引会将索引从数据库中删除。

可以使用 Sybase Central 或 Interactive SQL 执行此任务。

**Sybase Central**

在 Sybase Central 中，您可在使用所选数据库时执行此任务。

◆ **删除 UltraLite 索引 (Sybase Central)**

1. 连接到 UltraLite 数据库。
2. 在左窗格中，双击 [索引]。
3. 右击索引，然后选择 [删除]。
4. 单击 [是]。

**Interactive SQL**

在 Interactive SQL 中，删除 (Delete) 索引也称作将其删除 (Drop)。您可以使用 DROP INDEX 语句执行此任务。

◆ **删除 UltraLite 索引 (Interactive SQL)**

1. 连接到数据库。
2. 执行 DROP INDEX 语句。

例如，以下语句会从数据库中删除 EmployeeNames 索引：

```
DROP INDEX EmployeeNames;
```

**另请参见**

- [“UltraLite DROP INDEX 语句”一节第 445 页](#)

## 使用 UltraLite 发布

发布是一个数据库对象，它标识将要同步的数据。如果您想要同步 UltraLite 数据库中的所有表和这些表中的所有行，请不要创建任何发布。

发布由一组项目组成。每个项目或许是一个完整表，或许是表中的行。您可以使用 WHERE 子句定义该行集（Palm OS 上的 HotSync 除外）。

每个数据库均可以有多个发布，这取决于您所需的同步逻辑。例如，您可能想要为高优先级数据创建一个发布。用户可通过高速无线网络同步此类数据。由于无线网络可产生使用开销，因此，您可能只想将这些使用费投资于业务关键数据。可在以后再同步底座中所有其它不太紧急的数据。

可以使用 Sybase Central 或使用 CREATE PUBLICATION 语句创建发布。在 Sybase Central 中，所有发布和项目都出现在 [发布] 文件夹中。

### 用法说明

- UltraLite 发布不支持列子集定义，也不支持 SUBSCRIBE BY 子句。如果 UltraLite 表中的列与 SQL Anywhere 统一数据库中的表不完全匹配，可以使用 MobiLink 脚本解决这些差异。
- 发布决定选择哪些列，但不决定这些列的发送顺序。列始终是以它们在 CREATE TABLE 语句中定义的顺序发送。
- 在发布中不需要设置表同步顺序。如果表顺序对于您的部署很重要，可在同步 UltraLite 数据库时通过设置 [表顺序] 同步参数设置表顺序。
- 由于 UltraLite 不支持对象所有权，因此，任何用户均可删除发布。

### 另请参见

- “UltraLite 中的表顺序”一节第 126 页
- “发布数据”一节《MobiLink - 客户端管理》
- “设计 UltraLite 中的同步”一节第 123 页
- “同步脚本介绍”一节《MobiLink - 服务器管理》

## 在 UltraLite 中发布整个表

可创建的最简单发布仅包含单个项目，该项目由一个表的所有行和列组成。

可以使用 Sybase Central 或 Interactive SQL 执行此任务。

### Sybase Central

在 Sybase Central 中，您可在使用所连接的数据库时执行此任务。

#### ◆ 发布一个或多个完整 UltraLite 表 (Sybase Central)

1. 连接到 UltraLite 数据库。
2. 右击 [发布] 文件夹，然后选择 [新建] » [发布]。
3. 在 [您要给新发布指定什么名称] 字段中键入新发布的名称。单击 [下一步]。

4. 在 [表] 选项卡上, 从 [可用表] 列表中选择表。单击 [添加]。
5. 单击 [完成]。

## Interactive SQL

在 Interactive SQL 中, 您可以使用 CREATE PUBLICATION 语句执行此任务。

### ◆ 发布一个或多个完整 UltraLite 表 (Interactive SQL)

1. 连接到 UltraLite 数据库。
2. 执行一个 CREATE PUBLICATION 语句, 该语句指定新发布的名称和要发布的表。

例如, 以下语句创建一个发布, 该发布将发布整个 customer 表:

```
CREATE PUBLICATION pub_customer (  
    TABLE customer  
);
```

## 另请参见

- [“UltraLite CREATE PUBLICATION 语句”一节第 435 页](#)
- [“UltraLite 客户端”第 115 页](#)

## 从 UltraLite 表发布行的子集

发布只能包含特定的表行。在 Sybase Central 或 Interactive SQL 中, WHERE 子句会对那些上载的行加以限制, 这些行进行过更改并满足 WHERE 子句中的搜索条件。

若要上载所有更改的行, 则请不要指定 WHERE 子句。

### **Palm OS**

您不能在此平台上使用带有 WHERE 子句的 CREATE PUBLICATION 语句。

## Sybase Central

在 Sybase Central 中, 您可在使用所连接的数据库时执行此任务。

### ◆ 只发布 UltraLite 表中的某些行 (Sybase Central)

1. 连接到 UltraLite 数据库。
2. 右击 [发布] 文件夹, 然后选择 [新建] » [发布]。
3. 在 [您要给新发布指定什么名称] 字段中键入新发布的名称。
4. 单击 [下一步]。
5. 在 [可用表] 列表中, 选择表并单击 [添加]。

6. 单击 [**WHERE 子句**] 选项卡，并从 [**项目**] 列表中选择表。（可选步骤）可以使用 [**插入**] 窗口来协助对搜索条件进行格式设置。
7. 单击 [**完成**]。

## Interactive SQL

在 Interactive SQL 中，您可以使用 CREATE PUBLICATION 语句执行此任务。

### ◆ 使用 WHERE 子句在 UltraLite 中创建发布 (Interactive SQL)

1. 连接到 UltraLite 数据库。
2. 执行 CREATE PUBLICATION 语句，其中包括想要包含在发布中的表和一个 WHERE 条件。  
例如，以下示例创建一个单项目发布，其中包括编号为 856 的销售代表的所有销售订单信息：

```
CREATE PUBLICATION pub_orders_samuel_singer
( TABLE SalesOrders
  WHERE SalesRepresentative = 856 );
```

## 另请参见

- “UltraLite CREATE PUBLICATION 语句” 一节第 435 页
- “UltraLite 客户端” 第 115 页

## 删除 UltraLite 的发布

可使用 Sybase Central 或 Interactive SQL 删除发布。

## Sybase Central

在 Sybase Central 中，您可在使用所连接的数据库时执行此任务。

### ◆ 删除发布 (Sybase Central)

1. 连接到 UltraLite 数据库。
2. 在左窗格中，双击 [**发布**] 文件夹。
3. 右击发布，然后选择 [**删除**]。
4. 单击 [**是**]。

## Interactive SQL

在 Interactive SQL 中，删除 (Delete) 发布也称作将其删除 (Drop)。您可以使用 DROP PUBLICATION 语句执行此任务。

### ◆ 删除发布 (Interactive SQL)

1. 连接到 UltraLite 数据库。

2. 执行 DROP PUBLICATION 语句。

例如，以下语句将删除名为 `pub_orders` 的发布：

```
DROP PUBLICATION pub_orders;
```

**另请参见**

- [“UltraLite DROP PUBLICATION 语句” 一节第 446 页](#)
- [“UltraLite 客户端” 第 115 页](#)



## 使用 UltraLite 用户

UltraLite 数据库中对用户 ID 和口令进行了加密，因此，您只能查看 Sybase Central 中所定义的用户列表。

UltraLite 用户 ID 与 MobiLink 用户名或 SQL Anywhere 用户 ID 不同。

### 限制

创建唯一的用户 ID 时，请谨记以下限制：

- 针对每个数据库，UltraLite 最多可支持四个唯一用户。
- 用户 ID 和口令值最多只能包含 31 个字符。
- 口令始终区分大小写，而用户 ID 始终不区分大小写。您可随时通过 Sybase Central 更改口令。
- 用户 ID 中的任何前导和尾随空格都将被忽略。用户 ID 不能包含前导单引号 (')、前导双引号 (") 或分号 (;)。
- 用户 ID 一经创建，即不能对其进行更改。而是必须删除此用户 ID，然后添加一个新用户 ID。
- 可使用 Sybase Central 更改口令。

## 添加新 UltraLite 用户

UltraLite 不支持使用 Interactive SQL 创建用户。但是，您可以通过以下方法来添加用户：

- 使用 Sybase Central 将用户添加到 [用户] 文件夹中。
- 使用 Connection 对象上的 GrantConnectTo 函数从 UltraLite 应用程序中添加新用户。

### ◆ 创建新 UltraLite 用户 (Sybase Central)

1. 连接到 UltraLite 数据库。
2. 右击 [用户] 文件夹，然后选择 [新建] » [用户]。
3. 请按照向导中的说明进行操作。确保您了解 UltraLite 是如何解释不同的用户 ID 和口令组合的。请参见“解释用户 ID 和口令组合”一节第 44 页。

### 另请参见

- UltraLite.NET: “验证用户”一节 《UltraLite - .NET 编程》
- UltraLite C/C++: “验证用户”一节 《UltraLite - C 及 C++ 编程》
- UltraLite for M-Business Anywhere: “验证用户”一节 《UltraLite - M-Business Anywhere 编程》
- UltraLite for embedded SQL: “验证用户”一节 《UltraLite - C 及 C++ 编程》

## 删除现有 UltraLite 用户

UltraLite 不支持使用 SQL 语句删除用户。但是，您可以通过以下方法来删除用户：

- 使用 Sybase Central 从 [用户] 文件夹中删除用户。
- 使用 Connection 对象上的 RevokeConnectFrom 函数从 UltraLite 应用程序中删除用户。

### ◆ 删除现有 UltraLite 用户 (Sybase Central)

1. 连接到 UltraLite 数据库。
2. 在左窗格中，双击 [用户] 文件夹。
3. 右击用户，然后选择 [删除]。

### 另请参见

- UltraLite.NET: “验证用户”一节 《UltraLite - .NET 编程》
- UltraLite C/C++: “验证用户”一节 《UltraLite - C 及 C++ 编程》
- UltraLite for M-Business Anywhere: “验证用户”一节 《UltraLite - M-Business Anywhere 编程》
- UltraLite for embedded SQL: “验证用户”一节 《UltraLite - C 及 C++ 编程》

## 使用事件通知

UltraLite 现在支持事件和通知。通知是事件发生时发送的消息，还提供其它参数信息。UltraLite 具有系统事件，同时事件也可以是用户定义的。

事件通知使您可以在连接之间或连接到同一数据库的应用程序之间进行协调并发送信号。通知按队列管理：可以是按连接的缺省队列，也可以是按显示创建和命名的队列（可选）。发生事件时，会将通知发送到注册队列（或连接）。

每个连接管理自己的通知队列。可以为任意连接创建命名的队列。

使用预定义系统事件，此功能还可以为数据更改（例如对表进行更改时）提供 "触发器"，或者在发生同步时发送信号。预定义事件包括：

- 提交
- SyncComplete
- TableModified

也可以由应用程序来定义和触发用户事件。

为每种受支持的语言提供了事件和通知的 API。此外，还提供了一个 SQL 函数以访问 API 功能。

### 事件

事件	发生时间
<b>Commit</b>	完成提交后发信号时。
<b>SyncComplete</b>	完成同步后发信号时。
<b>TableModified</b>	插入、更新或删除表中的行时触发。此事件针对每个请求只发生一次，无论为此事件注册时多少行受到该请求的影响。  <i>object_name</i> 参数指定要监控的表。值为 "*" 表示数据库中的所有表。  <i>table_name</i> 通知参数是已修改表的名称。

```
note_info.event_name = "SyncComplete";
note_info.event_name_len = 12;
note_info.parms_type = ul_ev_note_info::P_NONE;

note_info.event_name = "TableModified";
note_info.event_name_len = 13;
note_info.parms_type = ul_ev_note_info::P_TABLE_NAME;
note_info.parms = table->name->data;
note_info.parms_len = table->name->len;
```

### 使用队列

可以创建和取消队列。

**CreateNotificationQueue** 为当前连接创建事件通知队列。队列名的使用范围仅限于每个连接，所以不同的连接可以创建具有相同名称的队列。发送事件通知时，数据库中具有匹配名称的所有队列都

收到单独的通知实例。名称不区分大小写。如果没指定任何队列，则按需为每个连接创建缺省队列。如果该连接的队列名称已存在或无效，则调用失败并出现错误。

**DestroyNotificationQueue** 取消给定事件通知队列。如果未读通知仍然在队列中，则会发出警告信号。放弃未读通知。如果已创建连接的缺省事件队列，则在连接关闭时会将其取消。

## 使用事件

**DeclareEvent** 声明稍后可以进行注册和触发的事件。UltraLite 预定义一些由对数据库或环境的操作触发的系统事件。事件名称必须唯一且名称不区分大小写。如果事件成功声明，则返回 **true**；如果名称已用过或无效，则返回 **false**。

**RegisterForEvent** 注册队列来接收事件的通知。如果不提供队列名称，则暗指采用缺省连接队列，如果需要，可以创建缺省连接队列。某些系统事件允许应用该事件的对象名称的规范。例如，**TableModified** 事件可以指定表名称。不像 **SendNotification**，仅特定的注册队列会收到事件的通知；不同连接中同名的其它队列则不会收到通知（除非它们也显示注册）。如果注册成功，则返回 **true**；如果队列或事件不存在，则返回 **false**。

**TriggerEvent** 触发事件并发送通知到所有注册的队列。返回发送的事件通知数。参数可能以 **name=value**；对提供。

## 使用通知

**SendNotification** 发送通知到数据库中匹配给定名称的所有队列（包括当前连接中任何此类队列）。此调用不会阻塞。使用特殊队列名称 "\*" 以发送到所有队列。返回发送的通知数（匹配队列的数目）。参数可能以 **name=value**；对提供。

**GetNotification** 读取事件通知。此调用将阻塞直到收到通知或给定等待时期到期。要取消等待，发送另一通知到给定队列或使用 **CancelGetNotification**。读取通知后，使用 **ReadNotificationParameter** 索引附加参数。如果事件已读取，则返回 **true**；如果等待时期到期或被取消，则返回 **false**。

**GetNotificationParameter** 获取刚刚由 **GetNotification** 读取的事件通知的已命名参数。给定队列中仅来自最近读取的通知的参数可用。如果找到参数，则返回 **true**；如果没找到参数，则返回 **false**。

**CancelGetNotification** 取消所有列中匹配给定名称的所有待执行的 **GetNotification** 调用。返回受影响的队列数（不一定是受阻塞读取的数目）。

## 其它注意事项

- 通知队列和事件名不能超过 32 个字符。
- 为控制系统资源，通知的数目也要限制。超过此限制时，**SQL\_E\_EVENT\_NOTIFICATION\_QUEUE\_FULL** 发出信号并放弃待执行通知。

## SQL 直通的 UltraLite 支持

SQL 直通功能允许统一数据库的发布者发送 SQL 语句脚本到 UltraLite 来执行。SQL 语句脚本可以自动执行也可以由具有 DBA 权限的用户手动执行。

同步期间，SQL 语句脚本自动下载到 UltraLite 并存储在 `sysssl` 表中。执行基于文件的下载或 Ping 时或者下载重新启动时，不会下载 SQL 语句脚本。

下次启动数据库时，SQL 语句脚本自动运行，除非：

- `flags` 参数在执行脚本中设置为 `manual`。
- 已设置了连接参数 `dont_run_scripts`。
- 上载失败。

如果 `flags` 参数在执行脚本中设置为 `manual`，可以使用以下方法来手动应用发送自统一数据库脚本：

- `ExecuteSQLPassthroughScripts`
- `ExecuteNextSQLPassthroughScript`
- `GetSQLPassthroughScriptCount`

同步期间，UltraLite 上载自上次同步后执行的所有脚本的状态。如果 UltraLite 在执行脚本时遇到问题，则不会进一步执行脚本直到通知统一数据库并发送继续执行的说明。已执行脚本的状态存储在统一数据库中的“`ml_passthrough_status`”一节《MobiLink - 服务器管理》表中。可以查看该表来确定分布式直通脚本是否成功。

可以使用观察器回调函数来监控自动或手动执行的脚本的进度。请参见“`ULRegisterSQLPassthroughCallback` 的回调函数”一节《UltraLite - C 及 C++ 编程》。

### 另请参见

- “SQL 直通简介”一节《MobiLink - 客户端管理》
- “`ExecuteSQLPassthroughScripts` 方法”一节《UltraLite - .NET 编程》
- “`ExecuteNextSQLPassthroughScript` 方法”一节《UltraLite - .NET 编程》
- “`GetSQLPassthroughScriptCount` 方法”一节《UltraLite - .NET 编程》

---

---

# UltraLite CustDB 示例

## 目录

CustDB 示例文件位置 .....	87
第 1 课：构建和运行 CustDB 应用程序 .....	89
第 2 课：登录和填充 UltraLite 远程数据库 .....	91
第 3 课：使用 CustDB 客户端应用程序 .....	92
第 4 课：与 CustDB 统一数据库同步 .....	94
第 5 课：浏览 MobiLink 同步脚本 .....	96
构建您自己的应用程序 .....	97

---

CustDB 示例由 SQL Anywhere 来安装。它是一种多层数据库管理解决方案，实现了 MobiLink 与 SQL Anywhere 统一数据库的同步。

CustDB 包含以下内容：

- 统一 SQL Anywhere 数据库。此数据库预填充了销售状态数据。
- 远程 UltraLite 数据库。此数据库初始为空。
- UltraLite 客户端应用程序。
- 带同步脚本的 MobiLink 服务器同步示例。

应用程序代码的不同版本为受支持的编程接口和平台而存在。但是，本教程只针对 Windows 桌面操作系统引用了应用程序的已编译版本。记住，各版本在实现 UltraLite 功能时为了遵循各自平台的约定，会稍有差异。

### 注意

一次只能运行一个 CustDB 实例。尝试运行多于一个实例会将第一个实例推到前台。

CustDB 允许销售人员跟踪和监控事务，然后收集来自两类用户的信息：

- 使用用户 ID 51、52 和 53 进行验证的销售人员。
- 使用用户 ID 50 进行验证的移动经理。

可将这些不同用户收集的信息与统一数据库同步。

学完各课程后，您将了解如何执行以下操作：

- 运行 MobiLink 服务器以执行统一数据库和 UltraLite 远程数据库之间的数据同步。
- 使用 Sybase Central 浏览 UltraLite 远程数据库中的数据。
- 使用 UltraLite 命令行实用程序管理 UltraLite 数据库。

### 另请参见

- [“CustDB 示例文件位置”一节第 87 页](#)
- [“方案”一节 《MobiLink - 入门》](#)
- [“CustDB 示例中的用户”一节 《MobiLink - 入门》](#)
- [“CustDB 数据库中的表”一节 《MobiLink - 入门》](#)



## CustDB 示例文件位置

### SQL Anywhere CustDB 数据库

这是统一数据库。在安装过程中，将为此数据库创建一个名为 SQL Anywhere 11 CustDB 的 ODBC 数据源。

CustDB 安装取决于您要使用现有示例，还是重新创建新文件：

- 对于现有示例：*samples-dir\UltraLite\CustDB\custdb.db*
- 会消除同步到统一 *CustDB.db* 文件的更改，因此可以使用原始版本：*samples-dir\UltraLite\CustDB\newdb.bat*

有关此文件的模式的详细信息，请参见“研究 MobiLink 的 CustDB 示例”《MobiLink - 入门》。

有关不同操作系统的 *samples-dir* 缺省位置的详细信息，请参见“示例目录”一节《SQL Anywhere 服务器 - 数据库管理》。

### UltraLite CustDB 数据库

只包含一个信息子集的统一数据库的远程版本，取决于同步此数据库的用户。

文件名和位置可因平台、编程语言，甚至是设备的不同而变化。

- 对于 UltraLite.NET：*samples-dir\UltraLite.NET\CustDB\Common\*
- 对于所有其它平台和 API：*samples-dir\UltraLite\CustDB\custdb.udb*

有关不同操作系统的 *samples-dir* 缺省位置的详细信息，请参见“示例目录”一节《SQL Anywhere 服务器 - 数据库管理》。

### RDBMS 特定的构建脚本

为受支持的 RDBMS 之一重新构建 CustDB 统一数据库的 SQL 脚本。

在 *samples-dir\MobiLink\CustDB* 目录中，您可以找到以下文件：

- 对于 SQL Anywhere：*custdb.sql*
- 对于 Adaptive Server Enterprise：*custase.sql*
- 对于 Microsoft SQL Server：*custmss.sql*
- 对于 Oracle：*custora.sql*
- 对于 IBM DB2：*custdb2.sql*

有关设置统一数据库的详细信息，请参见“建立 CustDB 统一数据库”一节《MobiLink - 入门》。

有关不同操作系统的 *samples-dir* 缺省位置的详细信息，请参见“示例目录”一节《SQL Anywhere 服务器 - 数据库管理》。

### UltraLite CustDB 客户端应用程序和自述文件

这些是提供了 UltraLite 远程数据库的用户友好界面的最终用户工具。针对每个受支持的 API 都装有一个示例客户端。

每个客户端应用程序还包括一个 *ReadMe.html* 或 *ReadMe.txt* 文件。这些文件的内容各异。但是在某些情况下，它们都概述了构建和运行该示例所需的步骤。

应用程序及其自述文件的位置视开发环境而定。请参见“[第 1 课：构建和运行 CustDB 应用程序](#)”一节第 89 页。

### SQL 同步逻辑

这些是查询和修改 UltraLite 数据库中的信息所需的 SQL 语句，以及启动与统一数据库的同步所需的调用。

*samples-dir\UltraLite\CustDB\custdb.sqc*

有关不同操作系统的 *samples-dir* 缺省位置的详细信息，请参见“[示例目录](#)”一节《[SQL Anywhere 服务器 - 数据库管理](#)》。

### 另请参见

- “[MobiLink 统一数据库](#)” 《[MobiLink - 服务器管理](#)》

# 第 1 课：构建和运行 CustDB 应用程序

针对多种开发环境构建 CustDB 应用程序。有关适用于全部环境的一般过程，请参见以下部分。

## ◆ 构建和运行 CustDB 应用程序

1. 构建 CustDB 应用程序：
  - a. 在适当环境中打开 CustDB 项目文件。
  - b. 编译源代码。
2. 运行 CustDB 应用程序：
  - a. 将 CustDB 可执行文件部署到移动设备上。
  - b. 将 UltraLite CustDB 数据库部署到移动设备上。
  - c. 运行 CustDB 可执行文件。

## 用于 Windows 32 位桌面操作系统的 UltraLite

运行前无需构建 CustDB 应用程序。

您可以在 `install-dir\UltraLite\win32\386` 目录下找到 CustDB 可执行文件。

## UltraLite for C/C++

- **C/C++ 的全部版本** 由于存在多个 C/C++ 开发环境，所以您可以找到 C/C++ CustDB 项目文件的多个版本。大多数版本使用通用文件。这些文件位于 `samples-dir\UltraLite\Custdb` 目录下。  
有关 C/C++ CustDB 应用程序所有版本的信息，请参见 `samples-dir\UltraLite\Custdb\readme.txt`。
- **CodeWarrior for Palm OS** 可以在 `samples-dir\UltraLite\CustDB\cwcommon` 和 `samples-dir\UltraLite\CustDB\cw` 目录中找到项目文件。  
有关构建 C/C++ CustDB 应用程序的详细信息，请参见“在 CodeWarrior 中构建 CustDB 示例应用程序”一节《UltraLite - C 及 C++ 编程》。
- **Visual Studio** 根据您的 Visual Studio 版本，可以在 `samples-dir\UltraLite\CustDB\vs9` 和 `samples-dir\UltraLite\CustDB\vs8` 目录中找到项目文件。要构建和运行 CustDB 应用程序，请遵循本课开始部分中给出的说明。

## UltraLite for Embedded SQL

根据您的 embedded Visual C++ 版本，可以在 `samples-dir\UltraLite\CustDB\EVC` 和 `samples-dir\UltraLite\CustDB\EVC40` 目录中找到特定于 eMbedded Visual C++ 的项目文件。

有关使用 eMbedded Visual C++ 为 Windows Mobile 构建嵌入式 SQL CustDB 应用程序的详细信息，请参见“构建 CustDB 示例应用程序”一节《UltraLite - C 及 C++ 编程》。

## UltraLite.NET

可以在 `samples-dir\UltraLite.NET\CustDB` 目录中找到特定于 Visual Studio.NET 的项目文件。

有关使用 Visual Studio.NET 为 Windows Mobile 构建 CustDB 应用程序的说明，请参见 *samples-dir\UltraLite.NET\CustDB\ce\ReadMe.html*。

要获取 Microsoft Windows 桌面操作系统的部署目录信息以及其它 UltraLite.NET 示例下载位置的信息，请参见 *samples-dir\UltraLite.NET\CustDB\Desktop\ReadMe.html*。

### UltraLite for M-Business Anywhere

可以在 *samples-dir\UltraLiteForMBusinessAnywhere\CustDB* 目录中找到特定于 M-Business Anywhere 的项目文件。

有关使用 M-Business Anywhere 构建 CustDB 应用程序的详细信息，请参见“[UltraLite for M-Business Anywhere 快速入门](#)”一节《[UltraLite - M-Business Anywhere 编程](#)》。说明适用于 Windows Mobile、Windows 和 Palm OS。

## 第 2 课：登录和填充 UltraLite 远程数据库

本课演示了如何：

- 启动示例 MobiLink 服务器。
- 启动示例 UltraLite 客户端应用程序。
- 登录到 UltraLite。

在本教程中，示例应用程序与 MobiLink 服务器运行在同一台式计算机上。但您可以将客户端应用程序部署到设备并获得相同的结果。

### ◆ 启动并同步示例应用程序

1. 选择 [开始] » [程序] » [SQL Anywhere 11] » [MobiLink] » [同步服务器示例]。或者，执行以下命令：

```
m1srv11 -c "DSN=SQL Anywhere 11 CustDB" -zu+ -vcrs
```

出现命令提示符来显示有关 MobiLink 服务器状态的消息。

2. 选择 [开始] » [程序] » [SQL Anywhere 11] » [UltraLite] » [Windows 示例应用程序]。
3. 在 [Employee ID] 字段键入 50。按 Enter 键。

应用程序进行同步，并且 MobiLink 服务器消息窗口将显示消息来指示同步正在进行。

缺省同步脚本会确定当用户 50 登录时要下载到应用程序的客户、产品和订单的子集。在本示例中，只下载尚未批准的订单。

4. 确认公司名称和示例订单出现在应用程序窗口中。

## 第 3 课：使用 CustDB 客户端应用程序

统一数据库和远程数据库都包含名为 ULOrder 的表。虽然统一数据库会保存全部订单（已批准的和等待批准的），但 UltraLite 远程数据库只根据已验证用户显示列的子集。

在客户端应用程序中，表中的列作为字段显示。当添加订单时，必须填充 Customer、Product、Quantity、Price 和 Discount 字段。您还可以追加其它详细信息，如 Status 或 Notes。时间戳列标识是否需要同步。

### ◆ 浏览订单

对于 UltraLite 客户端应用程序的各个版本，浏览订单的方法类似。

通过浏览订单，您将滚动通过本地 UltraLite 数据库中的数据。因为客户是按字母顺序排序的，所以您可以轻松地在列表中滚动并按姓名查找客户。

1. 要向下滚动客户列表，单击 [下一步]。
2. 要向上滚动客户列表，单击 [上一步]。

### ◆ 添加订单：

在 UltraLite 客户端应用程序的各种版本中，添加订单的方法类似。

通过添加订单，您修改了本地 UltraLite 数据库中的数据。进行同步时这些数据才会与统一数据库共享。

1. 选择 [Order] » [New]。
2. 在 [Customer] 列表中，选择 [Basements R Us]。
3. 在 [Product] 列表中，选择 [Screwmaster Drill]。此项的价格将自动输入到 [Price] 字段中。
4. 在 [Quantity] 字段中键入 20。
5. 在 [Discount] 字段中，键入 5（百分比）并单击 [OK]。

### ◆ 批准、拒绝和删除订单

因为您已作为 ID 50 验证过标识，所以您是一个能够与销售人员执行全部相同任务的经理，但同时您还额外具有接受或拒绝订单的能力。通过接受或拒绝订单，您将更改订单状态并添加供销售人员查看的附加说明。但是，在进行同步之前，统一数据库中的数据不会更改。

1. 批准 **Apple Street Builders** 的订单。
  - a. 要找到客户，单击 [Previous]。
  - b. 要批准订单，单击 [Order]，然后单击 [Approve]。
  - c. 在 [Note] 列表中，选择 [Good]。
  - d. 单击 [OK]。

该订单的状态显示为 [Approved]。
2. 拒绝 **Art's Renovations** 的订单。

- a. 转到列表中下一个订单，它来自 **Art's Renovations**。
- b. 要拒绝订单，单击 **[Order]**，然后单击 **[Deny]**。
- c. 在 **[Note]** 列表中，选择 **[Discount Is Too High]**。
- d. 单击 **[OK]**。

该订单的状态显示为 **[Denied]**。

3. 删除 **Awnings R Us** 的订单。

- a. 转到列表中的下一个订单，该订单来自 **Awnings R Us**。
- b. 通过选择 **[Order]** » **[Delete]** 删除此订单。

单击 **[OK]** 以确认删除。

此订单会标记为已删除。但在将更改同步到统一数据库之前，当前数据仍保留在 UltraLite 远程数据库中。

**另请参见**

- “CustDB 数据库中的表” 一节 《MobiLink - 入门》

## 第 4 课：与 CustDB 统一数据库同步

要进行同步，MobiLink 服务器必须正在运行。如果您已关闭 MobiLink 服务器，则需要重新启动该服务器。请参见“第 2 课：登录和填充 UltraLite 远程数据库”一节第 91 页。

示例应用程序的同步过程会从您的数据库中删除已批准的订单。

您可以使用 Interactive SQL 或 Sybase Central 连接到统一数据库来确认您的更改已得到同步。

### ◆ 同步 UltraLite 远程数据库

1. 要同步数据，从 [File] 菜单中选择 [Synchronize]。
2. 确认同步已发生。
  - 在远程数据库中，可以通过查看 **Apple Street Builders** 的已批准订单目前已删除，来确认全部所需事务已发生。通过浏览订单以确认此条目不存在来执行此操作。
  - 在统一数据库中，您还可以通过查看统一数据库中的数据来确认全部所需操作已发生。

### ◆ 确认同步 (Sybase Central)

1. 选择 [开始] » [程序] » [SQL Anywhere 11] » [Sybase Central]。
2. 选择 [连接] » [使用 SQL Anywhere 11 连接]。
3. 单击 [ODBC 数据源名称]。
4. 单击 [浏览]，然后选择 [SQL Anywhere 11 CustDB]。
5. 单击 [确定]。
6. 单击 [确定]。
7. 双击 [表]。
8. 双击 [ULOrder (DBA)]。
9. 单击 [数据] 选项卡，验证订单 5100 已批准，订单 5101 被拒绝，订单 5102 已删除。

### ◆ 确认同步 (Interactive SQL)

1. 从 Interactive SQL 连接到统一数据库。
  - a. 从 [开始] 菜单中选择 [程序] » [SQL Anywhere 11] » [Interactive SQL]。
  - b. 单击 [ODBC 数据源名称]，然后选择 [SQL Anywhere 11 CustDB]。
2. 要确认批准和拒绝已同步，执行以下语句：

```
SELECT order_id, status
FROM ULOrder
WHERE status IS NOT NULL;
```

结果显示订单 5100 已批准，5101 被拒绝。

3. 删除的订单的 order\_id 为 5102。下面的查询不返回任何行，这说明该订单已从系统中删除。



```
SELECT *  
FROM UOrder  
WHERE order_id = 5102;
```

## 第 5 课：浏览 MobiLink 同步脚本

CustDB 的同步逻辑作为 MobiLink 同步脚本保存在统一数据库中。同步逻辑允许您确定需要下载和/或上载统一数据库的哪些部分。您可以使用诸如基于时间戳的同步或快照同步之类的技术下载完整表或部分表（带行或列的子集）。

除了表、用户和发布外，您还可以使用 Sybase Central 浏览统一数据库中存储的同步脚本。Sybase Central 是向数据库中添加这些脚本的主要工具。

通过调用 `ml_add_connection_script` 或 `ml_add_table_script`，`custdb.sql` 文件会将每个同步脚本都添加到统一数据库中。连接脚本控制与特定表无关联的高级别事件。可以使用这些事件执行在每个同步期间所需的全局任务。表脚本允许在发生与特定表的同步相关的特定事件时执行操作，例如开始与结束行的上载、解决冲突和选择要下载的行等。

有关 CustDB 中使用的同步逻辑的详细信息，请参见“同步逻辑源代码”一节《MobiLink - 入门》。

有关在 CustDB 中实现同步的详细信息，请参见“同步设计”一节《MobiLink - 入门》。

### ◆ 浏览同步脚本

1. 从 [开始] 菜单中选择 [程序] » [SQL Anywhere 11] » [Sybase Central]。
2. 在 [连接] 菜单中，选择 [使用 MobiLink 11 连接]。
3. 单击 [ODBC 数据源名称]。
4. 单击 [浏览]，然后选择 [SQL Anywhere 11 CustDB]。
5. 单击 [确定]。
6. 单击 [确定]。
7. 双击 [连接脚本]。

右窗格中将列出一组同步脚本以及与这些脚本关联的一组事件。当 MobiLink 服务器执行同步过程时，它会触发一系列事件。此时将运行与其中的事件相关联的所有同步脚本。通过编写同步脚本并将它们指派给同步事件，您就可以控制在同步过程中执行的操作。

8. 单击 [同步表]。
9. 在右窗格中，双击 [ULCustomer]。

将显示特定于表的一组脚本及其相应的事件。这些脚本可控制 ULCustomer 表中的数据与远程数据库同步的方式。

### 另请参见

- “编写同步脚本” 《MobiLink - 服务器管理》
- “UltraLite 客户端” 第 115 页
- “连接脚本” 一节 《MobiLink - 服务器管理》
- “表脚本” 一节 《MobiLink - 服务器管理》

## 构建您自己的应用程序

使用受支持的接口之一来构建您自己的应用程序。有关详细信息，请参见：

- UltraLite C++: “教程：使用 C++ API 构建应用程序” 《UltraLite - C 及 C++ 编程》
- UltraLite.NET: “教程：构建 UltraLite.NET 应用程序” 《UltraLite - .NET 编程》
- UltraLite for M-Business Anywhere: “教程：M-Business Anywhere 示例应用程序” 《UltraLite - M-Business Anywhere 编程》

---

---

# UltraLite 性能与优化

## 目录

使用索引扫描 .....	100
确定优化程序使用的访问方法 .....	102
使用索引散列调整查询性能 .....	103
选择最佳的散列大小 .....	105
设置最大散列大小 .....	107
管理临时表 .....	108
刷新单个或分组的事务 .....	110
数据库的加密和模糊处理怎样影响性能 .....	111
UltraLite 优化策略 .....	112

---

UltraLite 会提供良好的 SQL 查询性能。索引扫描、直接页扫描和临时表是内部优化技术，可帮助您获得产品的最佳性能。但是，您可以根据运行的任何查询性能测试的结果，进一步调整这些功能。

## 使用索引扫描

索引是一组指向表行的指针，基于一个或多个表列中数据值的顺序。索引是数据库对象。索引创建之后由 UltraLite 自动维护。可以创建一个或多个索引以提高查询性能，或者根据创建的索引类型，以确保行值保持唯一。

索引根据部分或所有列中的值对表的行进行排序。创建索引时，您选择要索引的列的顺序会成为列实际出现在索引中的顺序。因此，有策略地使用索引时，可以大大地提高索引列上的搜索性能。

UltraLite 支持以下索引。这些索引可以是单列索引或多列索引（也称为复合索引）。无法为 LONG VARCHAR 或 LONG BINARY 列创建索引。

索引	特性
主键	必需。唯一键的一个实例。只能有一个主键。索引列中的值必须唯一且不能为 NULL。
外键 <sup>1</sup>	可选。索引列中的值可以重复。空性取决于创建列时是否允许 NULL 值。外键列中的值必须存在于被引用的表中
唯一键 <sup>2</sup>	可选。索引列中的值必须唯一且不能为 NULL。
非唯一索引	可选。索引列中的值可以重复并且可以为 NULL。
唯一索引	可选。索引列中的值不能重复但可以为 NULL。

<sup>1</sup> 外键可以引用主键或唯一键。

<sup>2</sup> 也称为唯一约束。

### 性能提示

- 在满足以下条件的任何列上创建索引：
  - 经常搜索列中的值
  - 查询使用列连接表
  - 列经常用于 ORDER BY、GROUP BY 或 WHERE 子句中
- 创建复合索引时，索引的第一列应为查询中的谓语最常用到的列。
- 确保索引不会过高地增加设备内存方面的更新维护开销。
- 不要创建或维护不必要的索引：修改列中的数据时必须更新索引，因此也会对索引执行所有插入、更新和删除操作。
- 在大表上创建索引。
- 不要创建冗余索引。例如，如果您在表 T 上用列 (x,y) 创建索引，如果表 T 上已经存在使用列 (x, y, z) 创建的另一个索引，则您创建了冗余索引。

**另请参见**

- “管理临时表”一节第 108 页
- “使用直接页扫描”一节第 108 页
- “查看 UltraLite 执行计划”一节第 320 页
- “关于复合索引”一节第 71 页
- “EXPLANATION 函数 [Miscellaneous]”一节第 359 页
- UltraLite C++: “GetPlan 函数”一节 《UltraLite - C 及 C++ 编程》
- UltraLite for M-Business Anywhere: “getPlan 方法”一节 《UltraLite - M-Business Anywhere 编程》
- “UltraLite page\_size 创建参数”一节第 182 页

## 确定优化程序使用的访问方法

为查询优化选择索引时，UltraLite 优化程序使用复杂的优化策略。除了简单查询之外，您无法轻易预先确定优化程序用来优化查询性能的索引，或是否使用了某个索引。复杂程度增加时，选择的索引取决于查询所需的子句。在某些情况下，FOR READ ONLY 子句的出现可能导致优化程序选择直接表扫描，而不是选择能提供更好查询性能的索引。

优化查询时，优化程序将查看查询的要求，并检查是否存在可用来提高性能的任何索引。如果使用任何索引都无法提高性能，优化程序将不扫描索引，而是使用临时表或直接页扫描。因此，您可能需要对索引进行试验并经常检查生成的执行计划以确保：

- 不是在维护优化程序不使用的索引。
- 将要创建的临时表的数目降到最低。请参见“[管理临时表](#)”一节第 108 页。

对于复杂查询，更很难预知要使用哪个索引。例如，如果查询包含 WHERE 谓语句，且除了 ORDER BY 子句以外还包含 GROUP BY 子句，则单独的一个索引可能不会满足此查询的搜索条件。因此，如果您创建了一个索引来满足 WHERE 谓语句的选择性要求，您可能会发现优化程序实际上并未使用此索引。相反，优化程序可能使用一个为 ORDER BY 条件提供更好性能的索引，因为可能需要更多地处理此子句。

### 检查执行计划

可以通过适当的 API 调用以编程方式检查执行计划，也可以在 Interactive SQL 中的 [计划查看器] 上检查执行计划。

- **如果未使用索引** 执行计划以如下形式出现：

```
scan(T)
```

- **如果使用临时表** 执行计划以如下形式出现：

```
temp [scan(T)]
```

- **如果使用索引** 执行计划包含索引名：

```
scan (T, index_name)
```



## 使用索引散列调整查询性能

可以通过选择最大**散列**的特定大小来调整查询的性能。散列键表示索引列的实际值。索引散列键旨在避免进行先查找行、后装载行然后再将行解出才能得出索引值的高开销操作。它通过将足够多的行实际数据与其行 ID 包括在一起来阻止这些操作。

UltraLite 通过行 ID 在数据库文件中定位实际行数据。如果将散列大小设置为 0（即禁用索引散列），则索引条目只包含此行的 ID。如果将散列大小设置为 0 以外的任意值，则仍使用散列键。散列键可以包含该行的所有或部分转换的数据，并与行 ID 一起存储在索引页中。

如何确定散列键包含行数据的多少：

- 部分由您配置的最大散列大小属性决定。请参见“[选择最佳的散列大小](#)”一节第 105 页。
- 部分由列数据类型的实际需求量决定。

### 一个散列示例

索引散列的值维护索引列实际行数据的顺序。例如，如果对 Employees 表的 LastName 列进行索引，您可以看到四个名字按以下顺序排列：

Anders

Anderseck

Andersen

Anderson

如果散列前六个字母，这些行值的散列键将按以下形式出现：

Anders

Anders

Anders

Anders

虽然这些条目看起来相同，但请注意列表中的第一个 Anders 用来表示 **Anders** 的实际行值。而列表中的最后一个 Anders 用来表示 **Anderson** 的实际行值。

现在，请看以下语句：

```
SELECT *  
FROM Employees  
WHERE LastName = 'Andersen';
```

如果 Employees 表只包含很高比例的类似于 Andersen 的名字，那么散列键可能不会提供足够的唯一性来获得任何性能优势。在这种情况下，UltraLite 无法确定是否有某一散列键实际满足此语句的条件。当复制的索引散列键存在时，UltraLite 仍然需要：

1. 查找与所述行 ID 匹配的表行。
2. 装载数据再解开数据，以便计算值。

仅当 UltraLite 可以辨别一定比例的唯一散列，从而直接对索引本身进行查询条件求值时，才能体现出性能优势。例如，如果 **Employees** 表有数千个名字，则通过六个字母的散列仍能获得足够的益处。但是，如果 **Employees** 表只包含过多的以 **Anders\*** 开始的名字，则应至少散列七个字母，以便唯一键的程度增大。所以，现在怎样用这些散列键表示此示例开始时的最初的四个人名。

Anders

Anderse

Anderse

Anderso

现在只需解开并计算四个行值中的两个，而不是全部四个。

### 另请参见

- [“UltraLite max\\_hash\\_size 创建参数”一节第 177 页](#)
- [“选择最佳的散列大小”一节第 105 页](#)
- [“UltraLite 性能与优化”第 99 页](#)
- [“添加 UltraLite 索引”一节第 72 页](#)

## 选择最佳的散列大小

所选择的 UltraLite 缺省最大散列大小（4 字节）适合于大多数部署。您可以增加大小以包含包括行 ID 在内的更多数据。但是，此更改可能增加索引的大小并将其分段存储在多个页面。结果，这种更改可能增加数据库的大小。增加最大散列大小的影响取决于表中的行数：例如，如果只有几行，大的索引散列键将仍适合索引页。在这种情况下，不会产生索引碎片。

当选择最佳散列大小时，应考虑数据类型、行数据和数据库的大小（尤其是表包含许多行时）。

确定您是否选择了最佳散列大小的唯一方法是在目标设备上针对 UltraLite 客户端应用程序运行基准测试。除了数据库大小自身的更改外，您需要观察不同散列大小如何影响应用程序和查询性能。

### 数据类型

如果要散列列中的全部值，请注意下表中各数据类型所需的大小。仅当 UltraLite 确实需要时，它才使用最大散列大小，并且从不超过指定的最大散列大小。如果列类型不使用全字节限制，UltraLite 始终使用较小的散列大小。

数据类型	用于散列整个值的字节
FLOAT、DOUBLE 和 REAL	不散列。
BIT 和 TINYINT	1
SMALL INT 和 SHORT	2
INTEGER、LONG 和 DATE	4
DATETIME、TIME、TIMESTAMP 和 BIG	8
CHAR 和 VARCHAR	若要散列整个字符串，最大散列大小（以字节为单位）必须与声明的列大小相匹配。在 UTF-8 编码的数据库中，始终将声明的大小乘以因子 2，但不超过允许的最大 32 字节。  例如，如果在非 UTF-8 编码的数据库中声明一个列为 VARCHAR(10)，则所需大小为 10 字节。但是，如果在 UTF-8 编码的数据库中声明同样的列，则用于散列整个字符串的大小为 20 字节。
BINARY	最大散列大小（以字节为单位）必须与声明的列大小相匹配。  例如，如果声明一个列为 BINARY(30)，则所需大小为 30 字节。
UUID	16

例如，如果为分别声明为 INTEGER 和 BINARY (20) 的两列复合索引设置 6 字节的最大散列大小，则基于数据类型大小需求，将发生以下情况：

- 散列 INTEGER 列中行的整个值并将其存储在索引中，因为散列整型数据只需要 4 字节。
- 只散列 BINARY 列的前 2 个字节并将其存储在索引中，因为 INTEGER 列使用了前 4 个字节。如果剩余的 2 字节不能散列 BINARY 列的适当数据量，则增加最大散列大小。

## 行数据

存储在数据库中的数据的行为也会影响散列索引的有效性。

例如，如果有一个在给定的列条目之间共享的公用前缀，则选择只能散列前缀的大小可能会使散列无效。在这种情况下，需要选择一个可确保不仅只能散列公共前缀的大小。如果公共前缀很长，应考虑根本不散列值。

如果非唯一索引存储许多重复值并且 UltraLite 无法散列整个值，散列可能无法提高性能。

## 数据库大小

每个索引页都具有一些固定的开销，但实际的索引条目使用了大部分的页空间。较大的散列大小表明每个索引条目较大，这意味着一页上能够容纳的条目较少。对于大表，具有大散列的索引比具有小散列或无散列的索引使用更多的页。需要更多页会增大数据库的大小，并降低性能。因为高速缓存只能保存固定数量的页面从而导致 UltraLite 交换页面，所以通常会出现性能降低。

下表给出了散列的大小如何影响存储索引中数据所需的页数的近似值。

表	页面大小	散列大小	条目数	所需页数
表 A	4 KB	0	1200	3 页
表 B	4 KB	32 个字节	116	3 页
表 C	4 KB	32 个字节	1200 个条目	11 页

## 另请参见

- [“UltraLite max\\_hash\\_size 创建参数” 一节第 177 页](#)
- [“UltraLite 性能与优化” 第 99 页](#)
- [“添加 UltraLite 索引” 一节第 72 页](#)
- [“UltraLite 中的数据类型” 一节第 290 页](#)

## 设置最大散列大小

可通过两种方法设置最大散列大小：

- 若要存储数据库的最大散列大小的缺省值，则在创建数据库时可以设置 `max_hash_size` 创建参数。如果不想以缺省的大小散列索引，将此值设置为 0。否则，可以将它更改为不超过 32 个字节的任何值，或保持 4 字节的 UltraLite 缺省值。
- 如果要替换缺省设置，可在创建新索引时设置指定的散列大小。执行下列操作之一：
  - 在 Sybase Central 中，创建新索引时设置 [最大散列大小] 属性。
  - 若使用 SQL，则在 CREATE TABLE 或 CREATE INDEX 语句中使用 WITH MAX HASH SIZE 子句。

### 另请参见

- [“UltraLite CREATE INDEX 语句”一节第 433 页](#)
- [“UltraLite CREATE TABLE 语句”一节第 440 页](#)

## 管理临时表

通常，优化程序会始终尝试避免创建临时表以返回查询结果，因为必须填充整个临时表后才能返回第一行。如果索引已存在，优化程序会尝试首先使用索引，然后作为最后一个手段仅创建临时表。

很难预料创建的索引是否可避免使用临时表。因此，应始终检查查询计划以确保 UltraLite 查询优化程序实际使用了创建的索引。

### 另请参见

- [“UltraLite 临时表”一节第 10 页](#)
- [“确定优化程序使用的访问方法”一节第 102 页](#)
- [“读取 UltraLite 执行计划”一节第 321 页](#)

## 使用直接页扫描

当直接从数据库页访问信息更有效时，作为索引扫描的替代方法 UltraLite 使用直接页扫描。只有当优化程序确认了以下内容时才使用直接页扫描：

- 预先存在的索引不能更有效地返回结果。
- 未使用查询执行更新。例如，已经将语句声明为 FOR READ ONLY（未指定 FOR 语句时的缺省设置），或者查询以数据明显无法更新的形式编写。

因为 UltraLite 直接从存储行的页面读取行，所以查询结果无顺序的返回。后继查询结果的顺序是不可预知的。如果需要行的顺序为可预知的且确定的，请使用 ORDER BY 子句在一致顺序中获取结果。反之，如果顺序并不重要，可以省略 ORDER BY 子句以提高查询性能。

### 注意

如果使用 Table API 编写应用程序，则无法使用直接页扫描。

可以查看 UltraLite 何时直接扫描页或使用哪个索引返回结果。请参见 [“确定优化程序使用的访问方法”一节第 102 页](#)。

## 恢复主键索引顺序

在 UltraLite 10.0.0 及之前的版本中，如果 UltraLite 优化程序未使用其它索引，则使用主键返回结果。因此，行根据主键索引的顺序进行排序。

如果结果必须按主键排序，应重新编写查询以包含 ORDER BY 子句。如果使用此子句排序行是不切实际的，可以考虑使用 ORDERED\_TABLE\_SCAN 连接参数。

### 提示

建议使用 ORDER BY 子句（只要有可能）。

**另请参见**

- “UltraLite ORDERED\_TABLE\_SCAN 连接参数 [不建议使用]” 一节第 219 页
- “使用索引扫描” 一节第 100 页
- “UltraLite SELECT 语句” 一节第 457 页

## 刷新单个或分组的事务

在 UltraLite 中可以通过延迟已提交事务刷新来选择恢复点。当 UltraLite 将提交发布到存储时，恢复点有助于控制事务中 SQL 语句的子集触发附加操作开销的时间。

缺省情况下，UltraLite 在提交后立即使用基于操作的缺省值将单个事务刷新到存储。对于某些部署，这些频繁的操作可能是多余的并限制了事务吞吐量。若要减少由此缺省方法引起的性能开销，可以选择基于状态的方法。尤其是对依赖自动提交操作的应用程序，此方法延迟将已提交事务刷新到存储所需的附加开销：

- **关于检查点** 可以设置自己的检查点，然后使用它发布该段时间执行的工作。在单个事务或多个事务中，可以根据需要使用任意多个检查点。
- **组合** 可以选择事务计数阈值和/或超时阈值发布执行的工作。

因为应用程序不需要等待来自 UltraLite 的响应，基于状态的延迟提交刷新可获得更好的性能和更干净的应用程序设计。通过对尚未完全完成其工作的数据进行更精细的控制，延迟提交刷新也可以将事务的暴露程度降至最低。例如，在销售应用程序中，在订单的所有条目被添加或甚至被批准之前，第二个应用程序可以使用该订单。

但是，考虑提交刷新已经延迟的事务的可恢复性是很重要的。无法恢复未发布的事务。因此，您需要在应用程序的数据完整性与其性能之间进行权衡。

### 另请参见

- “UltraLite COMMIT\_FLUSH 连接参数” 一节第 210 页
- “UltraLite commit\_flush\_count 选项 [临时性]” 一节第 202 页
- “UltraLite commit\_flush\_timeout 选项 [临时性]” 一节第 203 页
- “UltraLite CHECKPOINT 语句” 一节第 431 页
- UltraLite for embedded SQL: “ULCheckpoint 函数” 一节 《UltraLite - C 及 C++ 编程》
- UltraLite for C++: “Checkpoint 函数” 一节 《UltraLite - C 及 C++ 编程》



## 数据库的加密和模糊处理怎样影响性能

可以加密数据库以提高存储在 UltraLite 中的信息的安全性。然而，您应注意由于会有 5-10% 之间的开销增大，会导致性能的降低。对性能的确切影响取决于高速缓存的大小。如果高速缓存太小，加密会明显增大开销。但如果高速缓存足够大，您可能根本感觉不到任何差别。要确定对您的方案多大的高速缓存大小最佳，可以使用基准测试为数据库性能绘图。

### 对高速缓存施加负载

您可以对不同大小的高速缓存进行基准测试，然后观察性能的突然变化。您的高速缓存应足够大以拥有一组好的工作页。请考虑以下建议以帮助您对高速缓存施加负载：

- 在表上创建多个索引然后添加外键。
- 随机插入一些行（不是索引排序）。
- 创建较大的行，至少是数据库页面大小的 25%。
- 将索引散列设置为不为 0 的某个值。此增加的大小也会增大所需的访问页面。
- 基于最小的高速缓存大小开始为性能绘图。例如，Windows NT 为 256 KB（此平台所允许的最小高速缓存），或其它平台为 64 KB。

如果您发现增加高速缓存大小不能提高加密数据库的性能，那么应考虑将数据进行模糊处理而不是将数据进行加密。在仍可以提供一些安全方面优势的同时，模糊处理可以获得更好的性能；与高度加密相比，模糊处理算法使用较少的代码，所以执行了较少的计算。简单加密的性能基本上应仅比未加密的性能稍慢一些。但是，您的安全要求最终必须指定是否选择使用高度加密。

### 另请参见

- [“UltraLite 性能与优化” 第 99 页](#)
- [“UltraLite page\\_size 创建参数” 一节第 182 页](#)
- [“UltraLite fips 创建参数” 一节第 176 页](#)
- [“UltraLite CACHE\\_SIZE 连接参数” 一节第 208 页](#)
- [“UltraLite CREATE INDEX 语句” 一节第 433 页](#)

## UltraLite 优化策略

UltraLite 的不同 UltraLite 部署可能需要特定设置以根据您的需要调整数据库。

### Palm OS 的早期版本

Palm 设备的早期版本（例如，版本 4.x）只有大约 200 KB 的 RAM。由于 UltraLite 11 需要动态内存，此限制可能会引发问题。

**平台策略** 有两种方案：

- 升级到 Palm OS 5.x。此建议为首选选项。
- 将 UltraLite 降级到 9.0.x 版本。

**数据库优化策略** 应确保执行以下操作：

- **对于 CACHE\_SIZE 创建参数** 不要使用此参数建立一个连接，并允许 UltraLite 设置缺省值。如果设置自己的值，由于同步要求动态内存，可能无意间会引发问题。
- **对于 page\_size 创建数据库属性** 使用可能的最小页面大小为应用程序创建一个新数据库。虽然 1 KB 是最小的页面大小，Palm OS 通常使用 2 KB 的页面大小。

# UltraLite 用作 MobiLink 客户端

本节包含介绍如何设置和运行用于 MobiLink 同步的 UltraLite 客户端的材料。

---

UltraLite 客户端 .....	115
使用 UltraLite 的 ActiveSync 和 HotSync .....	131
UltraLite 同步参数和网络协议选项 .....	137



---

# UltraLite 客户端

## 目录

UltraLite 的内置同步功能 .....	116
自定义 UltraLite 客户端同步行为 .....	117
UltraLite 中的主键唯一性 .....	119
设计 UltraLite 中的同步 .....	123
使用 MobiLink 文件传输 .....	129

---

UltraLite 的运行时包含一个可将现场和流动员工与企业后台系统链接起来的内置双向同步框架，因此与其它远程客户端相比，更少涉及同步 UltraLite 数据。此内置框架意味着，在缺省情况下会自动同步 UltraLite 数据库中的所有数据。不熟悉 MobiLink 同步的用户可使用此缺省行为，除非业务需求要求必须通过自定义同步设计来变更要同步到统一数据库的 UltraLite 数据。

有关 UltraLite 的详细信息，请参见“[UltraLite 简介](#)”第 1 页。有关如何将 SQL Anywhere 数据库用作 MobiLink 客户端的信息，请参见“[SQL Anywhere 客户端](#)”《[MobiLink - 客户端管理](#)》。

### 提示

备份 UltraLite 应用程序的最佳方法是与统一数据库进行同步。要恢复 UltraLite 数据库，可先建立一个空数据库，然后通过同步从统一数据库填充该数据库。

### 提示

如果需要部署多个文件，或者需要将不同版本的文件定向到特定用户 ID，则可以考虑使用 MobiLink 服务器来传输文件。请参见“[使用 MobiLink 文件传输](#)”一节第 129 页。

## UltraLite 的内置同步功能

UltraLite 在其数据管理层中包含 MobiLink 同步技术。与 SQL Anywhere 远程不同，不必增加 UltraLite 资源占用量来包括同步功能。

内置于 UltraLite 运行时的重要同步功能包括行状态跟踪机制和进度计数器。

### 行状态跟踪机制

对于数据同步来说，跟踪表和行的状态特别重要。UltraLite 数据库中的每一行都有一个用于跟踪行状态的单字节标记。除了同步以外，UltraLite 还使用行状态来控制事务处理和数据恢复。请参见“[UltraLite 行状态](#)”一节第 12 页。

### 进度计数器

UltraLite 使用进度计数器来确保稳健的同步。为每个上载都指定一个唯一编号以对其进行标识。这使 UltraLite 可以在出现通信错误时确定上载是否成功。

首次创建新数据库时，UltraLite 始终将同步进度计数器设置为零。进度计数器的值为零时，会将数据库标识为新 UltraLite 数据库，这将指示 MobiLink 服务器为此客户端重设其状态信息。

#### **注意**

每发生一次同步，UltraLite 就会相应递增进度计数器的值，因此不能将一个 UltraLite 数据库同步到不同的统一数据库。如果进度计数器的值不为零并且与统一数据库中存储的序列号不匹配，MobiLink 同步会报告偏移不匹配，并且同步失败。

### 另请参见

- “[ml\\_subscription](#)”一节 《[MobiLink - 服务器管理](#)》

## 自定义 UltraLite 客户端同步行为

将自定义同步支持添加到 UltraLite 最多涉及三个任务：

- **在包含多个远程客户端的同步模型中维护主键唯一性** 必需。在同步系统中，主键是标识不同数据库（远程数据库和统一数据库）中相同行的唯一方式，也是检测冲突的唯一方式。因此，多个客户端必须遵守以下规则：

- 每个要同步的表都必须有一个主键。
- 绝不要更新主键的值。
- 主键在所有已同步的数据库中都必须是唯一的。

请参见“维护唯一主键”一节《MobiLink - 服务器管理》和“UltraLite 中的主键唯一性”一节第 119 页。

- **确保设置了日期列，使小数数据不会丢失** 对于 SQL Anywhere 统一数据库，这通常不是问题。但对于诸如 Oracle 的数据库，可能就需要考虑兼容性问题。例如，UltraLite 和 Oracle 数据库必须共享相同的时间戳精度。另外，还应该向 Oracle 数据库添加 TIMESTAMP，以避免 UltraLite 远程数据库向统一数据库上载数据时丢失小数秒数据。请参见“Oracle 统一数据库”一节《MobiLink - 服务器管理》和“UltraLite 精度创建参数”一节第 184 页。

- **说明要上载到统一数据库的数据子集** 可选。只有在不想按缺省设置同步所有数据的情况下，才需要执行此任务。要定向要同步的数据，可使用一个或多个子集技术。请参见“设计 UltraLite 中的同步”一节第 123 页。

例如，您可能想要为高优先级数据创建一个发布。用户可通过高速无线网络同步此类数据。由于无线网络可产生较高的使用开销，因此，您可能想将这些使用费投资于业务关键数据。可在以后再同步底座中不太紧急的数据。

- **从 UltraLite 应用程序初始化同步和提供描述会话的参数** 必需。编程同步包括两部分：描述会话，然后初始化同步操作。

描述会话主要涉及选择同步通信流（也称为网络协议）及其参数、设置同步脚本的版本和指派 MobiLink 用户。但也可以配置其它参数：例如，使用 `upload_only` 和 `download_only` 参数将缺省的双向同步更改为仅单向同步。请参见“向 UltraLite 应用程序添加同步”一节第 126 页。

所有其它重要的同步行为是通过 MobiLink 同步脚本控制的。您需要多个脚本，因为每个 MobiLink 远程数据库都可以包含统一数据库中的另一个数据子集。

其中包括：

- 哪些数据作为更新下载到 UltraLite 远程数据库中的表。
- 对从远程数据库上载的更改需要进行哪些处理。

这意味着您可以编写自己的同步脚本，以便以适合的方式在各远程数据库间进行数据分区。

**另请参见**

- “MobiLink 统一数据库” 《MobiLink - 服务器管理》
- “同步脚本介绍” 一节 《MobiLink - 服务器管理》
- “直接行处理” 《MobiLink - 服务器管理》
- “在远程数据库之间对行进行分区” 一节 《MobiLink - 服务器管理》



## UltraLite 中的主键唯一性

UltraLite 可使用 MobiLink 支持的任何技术维护主键唯一性。请参见“[维护唯一主键](#)”一节《[MobiLink - 服务器管理](#)》。

其中一个方法是使用全局 ID。全局 ID 会影响生成 GLOBAL AUTOINCREMENT 列值的方式。在 MobiLink 服务器必须管理多个客户端的同步时，您需要使用声明为 GLOBAL AUTOINCREMENT 的列。

GLOBAL AUTOINCREMENT 类似于 AUTOINCREMENT，只不过对域进行分区。UltraLite 仅从指派给数据库的全局 ID 的分区提供列值。

### 另请参见

- “[UltraLite global\\_database\\_id 选项](#)”一节第 204 页

## 在 UltraLite 中使用 GLOBAL AUTOINCREMENT

您可将 UltraLite 数据库中某列的缺省值声明为 GLOBAL AUTOINCREMENT 类型。但必须先为 UltraLite 远程数据库设置全局 ID 才能自动递增这些列 ID。

### 小心

通过 MobiLink 同步下载的 GLOBAL AUTOINCREMENT 列值不会更新 GLOBAL AUTOINCREMENT 值计数器。因此，如果一个 MobiLink 客户端将某值插入到另一客户端的分区，可能会发生错误。要避免此问题，请确保 UltraLite 应用程序的每个副本都仅将值插入到其自己的分区。

### ◆ 在 UltraLite 数据库中声明 GLOBAL AUTOINCREMENT 列

1. 为数据库的每个副本都指派一个唯一的全局 ID 号。

`global_database_id` 数据库选项在 UltraLite 数据库中设置该值。在部署 UltraLite 时，必须为每个数据库指派一个不同的标识号。请参见“[UltraLite global\\_database\\_id 选项](#)”一节第 204 页。

2. 允许 UltraLite 使用由 UltraLite 数据库编号唯一标识的分区向列提供缺省值。UltraLite 遵循以下规则：

- 如果列不包含当前分区的任何值，则第一个缺省值为  $pn + 1$ 。其中  $p$  代表分区大小， $n$  代表全局 ID 号。
- 如果列包含当前分区中的值，但全部小于  $p(n + 1)$ ，则下一个缺省值将为该范围内上一个最大值加 1。
- 缺省列值不受列中当前分区外的值的影响；也就是说，不受小于  $pn + 1$  或大于  $p(n + 1)$  的值的影响。如果通过 MobiLink 同步从另一个数据库中复制了这些值，则这些值就可能存在。

例如，如果为 UltraLite 数据库指派的全局 ID 为 1 并且分区大小为 1000，则将在范围 1001-2000 中选择该数据库的缺省值。如果为数据库另一副本指派的标识号为 2，则向同一列提供的缺省值的范围将为 2001-3000。

- 因为全局 ID 号不能设置为负值，所以 UltraLite 为 GLOBAL AUTOINCREMENT 列选择的值始终为正。最大标识号仅受列数据类型和分区大小的限制。
  - 如果未设置全局 ID 值，或者用尽分区中的值，则将 NULL 值插入列。如果不允许 NULL 值，则尝试插入行将会引起错误。
3. 如果已用尽或很快就要用尽声明为 GLOBAL AUTOINCREMENT 的列的可用值，则需要设置新的全局 ID。UltraLite 会从全局 ID 号所标识的分区中选择 GLOBAL AUTOINCREMENT 值，但只能选到最大值。如果超出这些值，UltraLite 便开始生成 NULL 值。通过指派新的全局 ID 号，UltraLite 可设置其它分区中相应的值。

选择新全局 ID 的一个方法是维护未用的全局 ID 值的池。该池的维护与主键池的维护方法相同。请参见“使用主键池”一节《SQL Remote》。

**提示**

UltraLite API 提供了获取已用编号的比例的方法。返回值是 0-100 范围内的 SHORT 类型数字，代表迄今为止已使用的值的百分比。例如，值为 99 表示几乎没有剩下尚未使用的值，应为数据库指派一个新标识号。设置此标识号的方法因使用的编程接口而异。

**另请参见**

- “替换自动增量列的分区大小”一节第 121 页
- “UltraLite global\_database\_id 选项”一节第 204 页
- UltraLite.NET: “Connection 属性”一节《UltraLite - .NET 编程》
- UltraLite C/C++: “Synchronize 函数”一节《UltraLite - C 及 C++ 编程》
- UltraLite for M-Business Anywhere: “setDatabaseID 方法”一节《UltraLite - M-Business Anywhere 编程》
- UltraLite for embedded SQL: “ULSetDatabaseID 函数”一节《UltraLite - C 及 C++ 编程》
- UltraLite.NET: “GlobalAutoIncrementUsage 属性”一节《UltraLite - .NET 编程》
- UltraLite C/C++: “GetSynchResult 函数”一节《UltraLite - C 及 C++ 编程》
- UltraLite for M-Business Anywhere: “getGlobalAutoIncrementUsage 方法”一节《UltraLite - M-Business Anywhere 编程》
- UltraLite for embedded SQL: “ULGlobalAutoincUsage 函数”一节《UltraLite - C 及 C++ 编程》

## 确定最近指派的 GLOBAL AUTOINCREMENT 值

您可检索在最近一次插入操作过程中选择的 GLOBAL AUTOINCREMENT 值。由于这些缺省值通常用于主键，所以了解生成的值可使您更便捷地插入引用第一行主键的行。可使用以下工具检查值：

- **UltraLite for C/C++** 使用 ULConnection 对象上的 GetLastIdentity 函数。请参见“GetLastIdentity 函数”一节《UltraLite - C 及 C++ 编程》。
- **UltraLite.NET** 使用 ULConnection 类上的 LastIdentity 属性。请参见“LastIdentity 属性”一节《UltraLite - .NET 编程》。
- **UltraLite for M-Business Anywhere** 使用 Connection 类上的 GetLastIdentity 方法。请参见“getLastIdentity 方法”一节《UltraLite - M-Business Anywhere 编程》。

返回值是数据库数据类型为 UNSIGNED BIGINT 的无符号 64 位整数。由于此语句仅确定最近指派的缺省值，所以您应在执行插入语句后尽快检索此值，以避免得到虚假结果。

**注意**

有时，单个 INSERT 语句可能会包含多个类型为 GLOBAL AUTOINCREMENT 的列。在这种情况下，返回值是生成的缺省值之一，但没有可靠的方法能确定是哪一个。因此，在设计数据库和编写 INSERT 语句时，应避免此情况。

## 替换自动增量列的分区大小

分区大小可以为任意正整数，但分区大小的选择一般需要保证任何一个分区内的编号资源不被用尽，或极少用尽。

对于 INT 或 UNSIGNED INT 类型的列，缺省分区大小是  $2^{16} = 65536$ ；对于其它类型的列，缺省分区大小是  $2^{32} = 4294967296$ 。因为这些缺省值可能不合适，所以最好明确指定分区大小。

一些数据类型的缺省分区大小在 UltraLite 应用程序中和在 SQL Anywhere 数据库中是不同的。如果要使不同数据库保持一致，则要显式声明分区大小。

### ◆ 替换 UltraLite 分区值 (Sybase Central)

1. 连接到 UltraLite 数据库。
2. 右击所选列并选择 [属性]。
3. 单击 [值] 选项卡。
4. 在 [分区大小] 字段中输入任何正整数。

### ◆ 在 UltraLite 中声明自动增量列 (Interactive SQL)

1. 连接到 UltraLite 数据库。
2. 执行包含 DEFAULT GLOBAL AUTOINCREMENT 子句的 CREATE TABLE 或 ALTER TABLE 语句，并在括号中指定分区大小。请参见“[UltraLite CREATE TABLE 语句](#)”一节第 440 页和“[UltraLite ALTER TABLE 语句](#)”一节第 428 页。

例如，下面的语句将创建一个简单引用表，其中包含两列：用于保存客户标识号的整数列和用于保存客户名称的字符串列。此表需要的分区大小为 5000。

```
CREATE TABLE customer (  
    id INT DEFAULT GLOBAL AUTOINCREMENT (5000),  
    name VARCHAR(128) NOT NULL,  
    PRIMARY KEY (id)  
);
```

### 另请参见

- UltraLite.NET: “[GetColumnPartitionSize 方法](#)”一节 《[UltraLite - .NET 编程](#)》
- UltraLite C/C++: “[GetGlobalAutoincPartitionSize 函数](#)”一节 《[UltraLite - C 及 C++ 编程](#)》
- UltraLite for M-Business Anywhere: “[getColumnPartitionSize 方法](#)”一节 《[UltraLite - M-Business Anywhere 编程](#)》
- UltraLite for embedded SQL: “[ULGlobalAutoincUsage 函数](#)”一节 《[UltraLite - C 及 C++ 编程](#)》

## 设计 UltraLite 中的同步

缺省情况下，将同步 UltraLite 数据库中的所有数据。如果您不了解如何将 UltraLite 部署为 MobiLink 远程数据库，最初可以计划同步整个 UltraLite 远程数据库。

当您熟悉此过程后，可以决定自定义同步操作的行为以捕获更复杂的业务逻辑。设计自定义同步行为时，您需要向自己提问以下问题。如果业务需求很简单，可能只需要使用一个单独的同步功能。但在非常复杂的部署中，可能需要使用多个同步功能，以便配置所需的同步行为。

设计问题	如果回答是，请遵照以下指导和说明
是否要从同步中排除某些表？	nosync 表名后缀可以标识任何不想进行同步的表。请参见“UltraLite 中的 nosync 表”一节第 124 页。
是否只想同步整个表（即使数据未更改）？	allsync 表名后缀允许同步整个表，即使没有检测到任何更改。请参见“UltraLite 中的 allsync 表”一节第 124 页。
是要同步整个表还是仅同步满足特定条件的行？是否有一些数据由于其重要性或时间敏感性而需要同步优先级？	一个发布可包含多个用于列出需要同步的表的项目。一个项目可包含一个 WHERE 子句，该子句可根据行是否满足定义的条件来指定要上载的行。  可用多个发布来解决某些 UltraLite 数据需要在其它数据之前上载的优先级问题。请参见“UltraLite 中的发布”一节第 124 页。
是否因为有外键循环而需要用于同步的表顺序？	[表顺序] 同步参数用于在有外键循环时确定同步操作的顺序。但通常不建议在 UltraLite 中使用外键循环。请参见“UltraLite 中的表顺序”一节第 126 页。
是否要控制同步行为？例如，是否需要下载和上载同时发生？还是要将双向同步更改为仅单向同步？	在以下各项中使用相应的同步参数： <ul style="list-style-type: none"> <li>● 应用程序的同步结构（或同步枚举）。</li> <li>● ulsync 实用程序的 -e 选项。</li> </ul> 请参见“UltraLite 同步参数和网络协议选项”第 137 页。
要将同步触发器设置为基于时间（也就是调度）、底座触发还是用户启动？还是需要上述几项的组合？	可以通过适当的接口以编程方式实现不同的行为。在某些情况下，HotSync 或 ActiveSync 可管理同步过程。请参见“向 UltraLite 应用程序添加同步”一节第 126 页。
是否希望 UltraLite 客户端启用 TLS？	您所选择的加密算法决定了您必须根据设备上运行的平台设置设备的方法。请参见“通过启用 TLS 的同步部署 UltraLite”一节第 51 页。

## 另请参见

- “同步过程”一节 《MobiLink - 入门》
- “上载和下载”一节 《MobiLink - 入门》

## UltraLite 中的 nosync 表

通过将 `_nosync` 后缀添加到表名，可控制何时从上载操作中排除整个表。可将这些非同步表用于统一数据库中不需要的客户端特定的持久数据。除了不进行同步外，这些表的使用方法和 UltraLite 数据库中其它表的使用方法完全相同。

如果创建带有 `_nosync` 后缀的表，只能在保留 `_nosync` 后缀的前提下重命名该表。例如，不允许使用以下带有重命名子句的 `ALTER TABLE` 语句，因为新名称不再以 `nosync` 作为结尾：

```
ALTER TABLE purchase_comments_nosync
RENAME comments;
```

若要纠正此问题，必须重写语句以包含此后缀：

```
ALTER TABLE purchase_comments_nosync
RENAME comments_nosync;
```

也可使用发布达到相同的效果。请参见“UltraLite 中的发布”一节第 124 页。

## UltraLite 中的 allsync 表

通过将 `_allsync` 后缀添加到表名，可控制是否更改上载期间的同步行为，以便同步所有表数据（即使自上次同步会话以来未更改任何数据）。

某些 UltraLite 应用程序需要可存储于 `allsync` 表中的用户/客户端特定的数据。因此，如果将该表中的数据上载到统一数据库中的临时表，则可使用这些数据来控制由其它脚本进行的同步，而无需在统一数据库中维护这些数据。例如，您可能希望 UltraLite 应用程序指出它们需要哪些通道或主题，并使用该信息下载相应的行。

## UltraLite 中的发布

发布定义了一组用于描述要同步的数据的项目。通常，每个项目可以是一个完整的表，也可以定义表中数据的一个子集。如果要定义一个由给定表中的行组成的子集，可包含一个可选谓词（`WHERE` 子句）。

发布比添加 `_nosync` 表后缀方法更灵活，并可以实现更精细的控制。若要分别同步 UltraLite 数据库的各个数据子集，请使用多个发布。然后将发布与仅上载或仅下载同步参数相结合，以有效地同步高优先级更改。

## 添加发布

可使用 Sybase Central 或从 Interactive SQL 将发布添加到 UltraLite 数据库。对于 UltraLite 同步，发布中的每个项目可能包含一个完整表，也可能包含一个 WHERE 子句（对于 Palm OS 上的 HotSync 除外）。

### 注意

UltraLite 发布不支持列子集定义，也不支持 SUBSCRIBE BY 子句。如果 UltraLite 表中的列与 SQL Anywhere 统一数据库中的表不完全匹配，可以使用 MobiLink 脚本解决这些差异。

在发布中不需要设置表同步顺序。如果表顺序对于您的部署很重要，可在同步 UltraLite 数据库时通过设置 [表顺序] 同步参数设置表顺序。

### ◆ 从 UltraLite 数据库发布数据 (Sybase Central)

1. 使用 UltraLite 插件连接到 UltraLite 数据库。
2. 右击 [发布] 文件夹，然后选择 [新建] >> [发布]。
3. 输入新发布的名称。单击 [下一步]。
4. 在 [表] 选项卡上，从 [匹配表] 列表中选择一表。单击 [添加]。此时，在右侧的 [所选表] 列表中就会显示该表。
5. 添加其它表。
6. 如果需要的话，单击 [Where] 选项卡指定要在发布中包含的行。但不能指定列的子集。如果使用 HotSync 同步，则不指定 WHERE 子句。
7. 单击 [完成]。

### ◆ 从 UltraLite 数据库发布数据 (Interactive SQL)

1. 连接到 UltraLite 数据库。
2. 执行一个 CREATE PUBLICATION 语句，该语句指定新发布的名称和要发布的表。

## 另请参见

- [“Download Only 同步参数”一节第 142 页](#)
- [“Upload Only 同步参数”一节第 157 页](#)
- [“Additional Parameters 同步参数”一节第 138 页](#)
- [“发布数据”一节《MobiLink - 客户端管理》](#)
- [“UltraLite CREATE PUBLICATION 语句”一节第 435 页](#)
- [“使用 UltraLite 发布”一节第 75 页](#)
- [“同步脚本介绍”一节《MobiLink - 服务器管理》](#)

## UltraLite 中的表顺序

通过设置 [表顺序] 同步参数，可控制同步操作的顺序。如果要为同步指定表顺序，可按编程方式或在测试期间作为 `ulsync` 实用程序的一部分使用 [表顺序] 参数。[表顺序] 参数用于指定要上载的表的顺序。请参见“[Additional Parameters 同步参数](#)”一节第 138 页。

只有在 UltraLite 数据库出现以下情况时，才需要显式设置表顺序：

- 有外键循环。然后必须列出循环中的所有表。
- 所具有的外键关系与统一数据库中所使用的不同。

### 避免与外键循环相关的同步问题

对于使用外键循环的 UltraLite 数据库，表顺序特别重要。将一系列表链接在一起构成闭环时，就发生了循环。但是，如果统一数据库和 UltraLite 远程数据库之间的循环不同，情况会变得很复杂，所以不建议使用外键循环。

使用外键循环时，应对表进行排序，使主表的操作在相关外表的操作之前发生。[表顺序] 参数确保了外表中的插入操作满足其外键参照完整性约束条件（对于删除之类的其它操作也是如此）。

除了表排序之外，在提交操作之前检查参照完整性也可以避免同步问题。如果统一数据库是 SQL Anywhere 数据库，则可将其中一个外键设置为 **check on commit**。这确保在提交阶段（而非启动操作时）就检查外键参照完整性。例如：

```
CREATE TABLE c (  
    id INTEGER NOT NULL PRIMARY KEY,  
    c_pk INTEGER NOT NULL  
);  
CREATE TABLE p (  
    pk INTEGER NOT NULL PRIMARY KEY,  
    c_id INTEGER NOT NULL,  
    FOREIGN KEY p_to_c (c_id) REFERENCES c(id)  
);  
ALTER TABLE c  
    ADD FOREIGN KEY c_to_p (c_pk)  
    REFERENCES p(pk)  
    CHECK ON COMMIT;
```

如果统一数据库由其他数据库供应商提供，则查看并确定该数据库是否有类似的检查参照完整性的方法。如果有，则应实现此方法。否则，就必须重新设计表关系以消除所有外键循环。

### 另请参见

- “[参照完整性与同步](#)”一节《[MobiLink - 入门](#)》

## 向 UltraLite 应用程序添加同步

在 UltraLite 中，同步首先打开通过配置的通信流（也称为网络协议）进行的与 MobiLink 服务器之间的特定连接。除了对直接网络连接的同步支持之外，Palm OS 设备还支持 HotSync 同步，Windows CE 设备还支持 ActiveSync 同步。



## 定义连接

与 MobiLink 服务器同步的每个 UltraLite 远程数据库都通过网络协议进行同步。使用同步流参数设置网络协议。支持的网络协议包括 TCP/IP、HTTP、HTTPS 和 TLS。对于所选择的协议，还需要提供用于定义其它必需连接信息（例如 MobiLink 服务器主机和端口）的流参数。

还必须通过 `user_name` 和版本参数提供 MobiLink 用户信息和同步脚本版本。

## 定义同步行为

可以通过设置不同的同步参数来控制同步行为。设置参数的方式取决于您所使用的特定 UltraLite 接口。

要考虑的重要行为包括：

- **同步方向** 缺省情况下，同步为双向。如果仅需要单向同步，则记得使用相应的 `upload_only` 或 `download_only` 参数。通过执行单向同步，可最大程度缩短所需的同步时间。另外，对于仅下载同步，不必在同步前提交对 UltraLite 数据库进行的所有更改。对同步中未涉及的表的未提交更改不会进行上载，因此，未完成的事务不会导致问题发生。

若要使用仅下载同步，必须确保不在本地更改与下载重叠的行。如果有任何数据进行了本地更改，则 UltraLite 应用程序中的同步将失败，并出现 `SQLE_DOWNLOAD_CONFLICT` 错误。

- **同步期间的并发更改** 在上载阶段，UltraLite 应用程序可以用只读方式访问 UltraLite 数据库。在下载阶段，允许进行读写访问，但如果应用程序更改了下载随后试图要更改的行，则下载将失败并回退。可以通过设置 `disable_concurrency` 同步参数来禁用同步期间对数据的并发访问。

### ◆ 向 UltraLite 应用程序添加同步代码

1. 以同步信息结构字段的形式为会话提供所需的同步参数和协议选项。

例如，使用 C/C++ API 时，通过在 `ul_synch_info` 结构中设置相应的值，向 UltraLite 应用程序添加同步：

```
ul_synch_info info;
    // define a sync structure named "info"
    ULEnableTcpipSynchronization( &sqlca );
    // use a TCP/IP stream
    conn->InitSynchInfo( &info );
    // initialize the structure
    info.stream = ULStream();
    // specify the Socket Stream
    info.stream_parms= UL_TEXT( "host=myMLserver;port=2439" );
    // set the MobiLink host information
    info.version = UL_TEXT( "custdb 11.0" );
    // set the MobiLink version information
    info.user_name = UL_TEXT( "50" );
    // set the MobiLink user name
    info.download_only =ul_true;
    // make the synchronization download-only
```

2. 初始化同步。

对于基于 TCP/IP 的直接同步，将调用 API 特定的同步函数。这些函数将返回一个布尔值，指明同步操作是成功还是失败。如果同步失败，可查看另一结构中的详细错误状态字段，以获取更多的错误信息。

对于 HotSync 同步，必须使用 ULSetSynchInfo 函数，从而以参数形式提供 ul\_synch\_info 结构。这将 ul\_synch\_info 结构附加到当前数据库，以供用于随后的同步。

对于 ActiveSync 同步，必须从 ActiveSync 提供程序捕获同步消息，并使用 DoSync 函数调用 ULSynchronize。

3. 如果要向用户报告同步进度，可使用观察器回调函数。

**提示**

如果您的环境中的 DLL 因为其本身太大或网络连接不可靠而失败，您可能要实现断点续传下载。请参见“处理失败的下载”一节《MobiLink - 服务器管理》和“恢复失败的下载”一节《MobiLink - 服务器管理》。

**另请参见**

- “使用 UltraLite 的 ActiveSync 和 HotSync” 第 131 页
- “仅上载同步和仅下载同步” 一节《MobiLink - 服务器管理》
- “上载和下载” 一节《MobiLink - 入门》
- “UltraLite 同步参数和网络协议选项” 第 137 页
- UltraLite.NET: “UltraLite 应用程序中的同步” 一节《UltraLite - .NET 编程》
- UltraLite C/C++: “同步数据” 一节《UltraLite - C 及 C++ 编程》
- UltraLite C/C++: “将 HotSync 同步添加到 Palm 应用程序” 一节《UltraLite - C 及 C++ 编程》
- UltraLite C/C++: “向应用程序添加 ActiveSync 同步” 一节《UltraLite - C 及 C++ 编程》
- UltraLite for M-Business Anywhere: “同步数据” 一节《UltraLite - M-Business Anywhere 编程》
- UltraLite for embedded SQL: “向应用程序添加同步” 一节《UltraLite - C 及 C++ 编程》

## 使用 MobiLink 文件传输

所有 UltraLite 库均支持使用 MobiLink 服务器传输文件的功能（M-Business Anywhere 除外）。M-Business Anywhere 不需要此功能，因为它有自己的文件部署或传输机制（称为**通道同步**）。

对于所有其它 API，在以下情况下使用 MobiLink 文件传输机制：

- 需要将多个文件部署到多个设备（特别是将公司防火墙用作安全措施时）。因为 MobiLink 已配置为通过这些防火墙处理同步，因此使用 MLFileTransfer 机制可以很方便地置备用于升级及其它类型文件传输的设备。
- 有文件要定向到特定的 MobiLink 用户 ID。这要求在 MobiLink 服务器上为需要的每个用户 ID 都创建一个或多个用户特定目录。如果只有单个版本的文件，则可使用缺省目录。

### 文件传输工作原理

您可以利用两个由 MobiLink 启动的文件传输机制之一将文件下载到设备：运行 mlfiletransfer 实用程序进行桌面传输，或调用适合于 UltraLite 应用程序编码所使用的 API 的函数。这两种方法都要求：

#### 1. 描述传输目标。

无论是从桌面操作系统使用 mlfiletransfer 实用程序，还是使用适合于 API 的函数，都必须设置目标设备或台式计算机上的文件的本地路径和文件名。如果应用程序中或最终用户未提供任何信息，则假定使用源文件名并将文件存储在当前目录中。

传输目标的目标目录因设备的操作系统而异：

- 在 Palm OS 上，如果目标是外部存储介质，则必须在本地路径目标前加上前缀 **vfs:**。  
如果目标为 NULL，则 mlfiletransfer 假定需要将 Palm 记录数据库文件（\*.pdb 文件）下载到设备的记录存储库。  
文件名必须遵循 Palm OS 的文件名约定。请参见“[Palm OS](#)”一节第 41 页。
- 在 Windows Mobile 上，如果目标为 NULL，则将文件存储在根目录（\）。  
文件名必须遵循 Windows Mobile 的文件名约定。请参见“[Windows Mobile](#)”一节第 41 页。
- 在桌面上，如果目标为 NULL，则将文件存储在当前目录。  
文件名必须遵循桌面系统的文件名约定。请参见“[Windows 桌面操作系统](#)”一节第 41 页。

#### 2. 设置用于标识用户和下载正确文件的用户证书。

此用户名和口令独立于任何数据库用户 ID 和口令，它们用于向 MobiLink 服务器标识和验证应用程序。

#### 3. 设置要使用的流类型，并定义所需流的参数。这些参数与 UltraLite 支持的 MobiLink 同步参数相同。请参见“[UltraLite 同步参数和网络协议选项](#)”第 137 页。

大多数同步流都需要用一些参数来标识 MobiLink 服务器地址和控制其它行为。如果设置的流类型值在平台上无效，则流类型将被设置为 TCP/IP。

#### 4. 描述传输机制所需的行为。

例如，可设置允许此机制强制进行下载（即使文件已存在于目标上并未进行更改）的属性，或设置允许继续完成部分下载的属性。也可设置是否想要监视和报告下载进度。

5. 确保 MobiLink 服务器正在运行，并且是使用 `-ftr` 选项启动的。
6. 启动传输，并在适用情况下监控下载进度。

通过显示下载进度，用户可取消下载并稍后恢复下载。

### 另请参见

- “`-ftr` 选项” 一节 《MobiLink - 服务器管理》
- “MobiLink 文件传输实用程序 (`mlfiletransfer`)” 一节 《MobiLink - 客户端管理》
- UltraLite for C/C++: “`MLFileTransfer` 函数” 一节 《UltraLite - C 及 C++ 编程》
- UltraLite.NET: “`ULFileTransfer` 类” 一节 《UltraLite - .NET 编程》
- UltraLite for M-Business: 不支持

---

# 使用 UltraLite 的 ActiveSync 和 HotSync

## 目录

Palm OS 上的 HotSync .....	132
Windows Mobile 上的 ActiveSync .....	135

---

## Palm OS 上的 HotSync

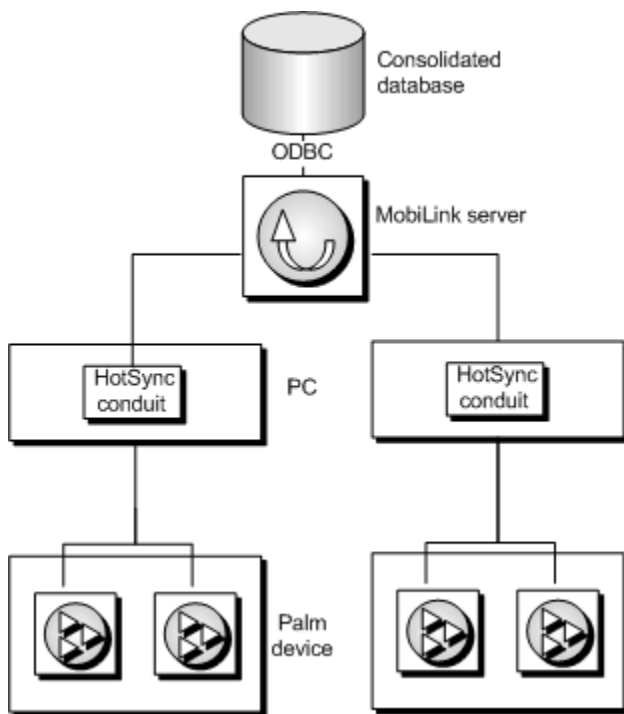
由于可在 Palm OS 设备上通过以太网、Wi-Fi 或 RAS 连接来同步数据，所以本节介绍如何为 HotSync 同步配置桌面操作系统和设备。

利用 HotSync 同步，可从用户的桌面操作系统同时管理 Palm 设备上所有 UltraLite 数据库的同步。当该设备连接到桌面时，HotSync 同步由 HotSync 从外部启动。这要求您对应用程序编程以进行 HotSync 同步。请参见“将 HotSync 同步添加到 Palm 应用程序”一节《UltraLite - C 及 C++ 编程》。

如果要从某个应用程序直接同步 UltraLite（即，不使用 Hotsync），则最终用户需要单独打开每个应用程序并依次同步每个数据库。通过使用 API 特定的同步函数初始化同步，以编程方式执行此操作。请参见“向 Palm 应用程序添加 TCP/IP、HTTP 或 HTTPS 同步”一节《UltraLite - C 及 C++ 编程》。

### HotSync 体系结构

HotSync 体系结构需要一个与远程数据库的传入和传出数据同步的统一数据库。MobiLink 服务器管理这些数据库间的同步事件。UltraLite Hotsync 管道从最终用户的桌面以本地方式管理同步事件。



必须使用 HotSync 管道安装实用程序 (ulcond11)，来安装管道和注册每一个 UltraLite 数据库。通过使用管道注册每个数据库，可将 UltraLite Hotsync 管道配置为：

- 管理所有与某个应用程序关联的数据库的 HotSync 同步（从而管理其注册的创建者 ID）。必须为每个适用的创建者 ID 安装管道。

- 使用相应的连接字符串连接到每个数据库。如果需要注册多个数据库，可通过 `ulcond11` 实用程序使用 `-a` 选项。这可将每个附加的连接字符串追加到使用 `-c` 选项定义的字符串上。

如此一来，您便可通过从桌面操作系统使用 HotSync 初始化同步，来同时同步所有数据库。

### 另请参见

- “Palm OS 的 UltraLite HotSync 管道安装实用程序 (`ulcond11`)” 一节第 243 页
- UltraLite C/C++: “开发用于 Palm OS 的 UltraLite 应用程序” 《UltraLite - C 及 C++ 编程》
- UltraLite for embedded SQL: “向应用程序添加同步” 一节 《UltraLite - C 及 C++ 编程》
- UltraLite for M-Business Anywhere: “同步数据” 一节 《UltraLite - M-Business Anywhere 编程》

## HotSync 同步概述

1. 确保已在具有 `ul_synch_info` 结构的应用程序中指定了必需的同步参数。然后，便可通过调用 `SetSynchInfo` 函数为 HotSync 同步设置这些同步参数。这些同步参数也包含与 MobiLink 服务器通信所需的网络协议选项。请参见“设置 MobiLink 同步的协议选项”一节第 133 页。
2. 关闭 UltraLite 应用程序时，UltraLite 应用程序的状态将存储在与该数据库分开的临时文件中。请参见“在 UltraLite Palm 应用程序中维护状态（不建议使用）”一节 《UltraLite - C 及 C++ 编程》。
3. 同步 Palm 设备时，HotSync 在桌面上为特定的创建者 ID 调用 UltraLite HotSync 管道。
4. UltraLite HotSync 管道通过使用相应的字符串连接到所有已注册的数据库，然后同步这些数据库。未正确调用 `SetSynchInfo` 的数据库将无法同步。
5. 在启动应用程序时，它将装载以前保存的 UltraLite 应用程序状态。请参见“在 UltraLite Palm 应用程序中恢复状态（不建议使用）”一节 《UltraLite - C 及 C++ 编程》。

### 另请参见

- “`ul_synch_info_a` 结构” 一节 《UltraLite - C 及 C++ 编程》
- “`SetSynchInfo` 函数” 一节 《UltraLite - C 及 C++ 编程》

## 设置 MobiLink 同步的协议选项

协议选项描述 UltraLite HotSync 管道与 MobiLink 服务器之间的连接。通常，要将此信息作为应用程序同步代码的一部分进行添加。但是，也可从 HotSync 甚至 `ulcond11` 输入必需的参数。

如果使用的是 `ul_synch_info` 结构，则该参数具有以下格式：

```
stream=name;parameters
```

`stream=name` 表示网络协议的类型。`stream=name` 的缺省设置为 TCP/IP。

`parameters` 是一组由管道使用的网络协议选项，它与正在使用协议的 `stream_parms` 参数具有相同的形式。对于给定的流，`parameters` 的缺省值与协议的 `stream_parms` 参数的缺省值相同。

**注意**

如果不提供协议选项，则管道将在注册表中搜索您运行 `ulcond11` 时也会提供的协议名和协议选项。如果 HotSync 未发现任何有效的网络协议，则将使用缺省的协议和协议选项。此缺省的流参数设置为：

```
stream=tcpip;host=localhost
```

**另请参见**

- “UltraLite 同步流的网络协议选项” 一节第 161 页
- UltraLite.NET: “ULStreamType 枚举” 一节 《UltraLite - .NET 编程》
- UltraLite for embedded SQL: “ULSetSynchInfo 函数” 一节 《UltraLite - C 及 C++ 编程》
- UltraLite for C/C++: “SetSynchInfo 函数” 一节 《UltraLite - C 及 C++ 编程》
- UltraLite for C/C++: “ul\_synch\_info\_a 结构” 一节 《UltraLite - C 及 C++ 编程》
- UltraLite for M-Business: “setStream 方法” 一节 《UltraLite - M-Business Anywhere 编程》

## 调试 HotSync 操作

HotSync 日志文件保存在 `Palm-install\User-dir` 目录中。缺省情况下，该文件包含以下信息：

- 同步事件的时间。
- 每个已注册管道的状态。

通过设置 `UL_DEBUG_CONDUIT_LOG` 环境变量，可获得该文件中的附加调试信息。根据需要捕获的信息量，有以下两个级别可供选择：

**UL\_DEBUG\_CONDUIT\_LOG = 1** 记录基本信息。例如同步参数和注册表位置。

**UL\_DEBUG\_CONDUIT\_LOG = 2** 记录附加的 UltraLite 详细信息，其中包括输入/输出操作。

重新启动 HotSync 以使新设置生效。

**另请参见**

- “SQL Anywhere 环境变量简介” 一节 《SQL Anywhere 服务器 - 数据库管理》



## Windows Mobile 上的 ActiveSync

由于可在 Windows Mobile 设备上通过以太网或 Wi-Fi 连接来同步数据，所以本节介绍如何将桌面操作系统和设备配置为使用 ActiveSync 同步。如果要使用其它替换方法之一直接同步，则需要对应用程序进行编程以使用相应的同步函数。

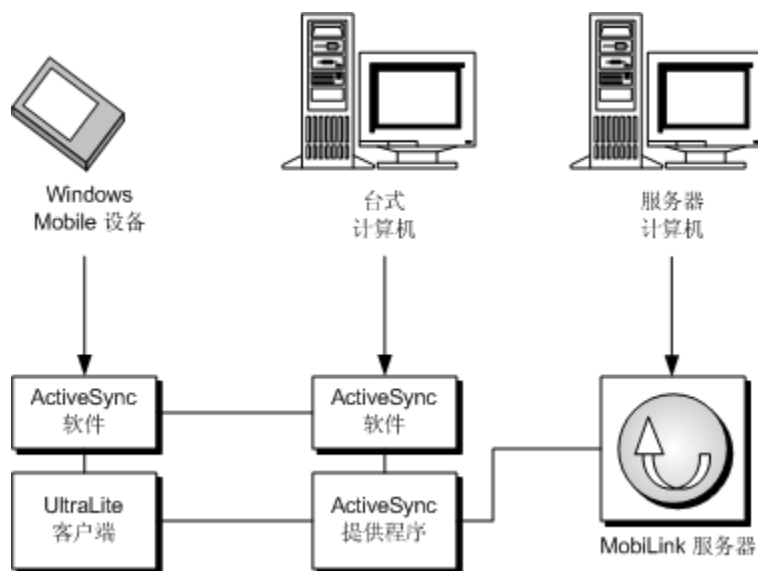
要使用 ActiveSync 启动的同步，需要：

- 通过 ActiveSync 注册所有需要使用 ActiveSync 启动的同步的应用程序。
- 将 ActiveSync 提供程序安装到桌面操作系统并将其部署到设备。

要确定哪些平台支持该提供程序，请参见 <http://www.sybase.com/detail?id=1062617>。

### ActiveSync 体系结构

下图显示 ActiveSync 体系结构需要的计算层。



请注意，除桌面操作系统外，您还必须在设备上安装 ActiveSync 提供程序。一台计算机只能有一个 ActiveSync 提供程序。但是，如果在一台 Windows Mobile 设备上安装了多个 UltraLite 应用程序，则可使用同一提供程序注册这些应用程序，以使其同时同步。

### 另请参见

- UltraLite C/C++: “向应用程序添加 ActiveSync 同步”一节 《UltraLite - C 及 C++ 编程》
- UltraLite for embedded SQL: “向应用程序添加同步”一节 《UltraLite - C 及 C++ 编程》
- UltraLite for M-Business Anywhere: “同步数据”一节 《UltraLite - M-Business Anywhere 编程》

## ActiveSync 同步概述

1. ActiveSync 开始同步会话。
2. ActiveSync 提供程序向该设备上第一个注册的应用程序发送同步通知消息。如果该应用程序尚未运行，则将其启动。
3. 为每个已注册的应用程序调用 WndProc。
4. 应用程序确定这是来自 ActiveSync 的同步通知消息后，该应用程序即会调用 ULIsSynchronizeMessage 以调用数据库同步过程。
5. 完成同步后，该应用程序即会调用 ULSignalSyncIsComplete 以通知提供程序它已完成同步。
6. 每个已使用提供程序注册的应用程序将重复执行步骤 2 至 5。

---

# UltraLite 同步参数和网络协议选项

## 目录

UltraLite 的同步参数 .....	138
UltraLite 同步流的网络协议选项 .....	161

---

## UltraLite 的同步参数

同步参数控制 UltraLite 数据库和 MobiLink 服务器之间的同步。设置参数的方式取决于您所使用的特定 UltraLite 接口。本章介绍这些参数的作用，并提供指向其它位置的链接，以便查看有关如何设置这些参数的信息。

### 注意

本章中介绍的参数仅适用于 UltraLite 远程数据库。要同步 SQL Anywhere 远程数据库，请参见“MobiLink SQL Anywhere 客户端实用程序 (dbmsync)”《MobiLink - 客户端管理》。

有关 API 特定的详细信息，请参见：

- UltraLite for Embedded SQL: “[ULSynchronize 函数](#)”一节《[UltraLite - C 及 C++ 编程](#)》
- UltraLite.NET: “[ULSyncParms 类](#)”一节《[UltraLite - .NET 编程](#)》
- UltraLite C/C++: “[Synchronize 函数](#)”一节《[UltraLite - C 及 C++ 编程](#)》
- UltraLite for M-Business Anywhere: “[SyncParms 类](#)”一节《[UltraLite - M-Business Anywhere 编程](#)》

### 必需参数

以下参数是必需的：

- **流类型** 请参见“[Stream Type 同步参数](#)”一节第 154 页。
- **用户名** 请参见“[User Name 同步参数](#)”一节第 158 页。
- **版本** 请参见“[Version 同步参数](#)”一节第 159 页。

如果未设置这些参数，同步函数将抛出异常（例如，SQLCode.SQLE\_SYNC\_INFO\_INVALID 或其等价的信息）。

### 有冲突的参数

只能指定以下参数之一：

- **仅下载** 请参见“[Download Only 同步参数](#)”一节第 142 页。
- **Ping** 请参见“[Ping 同步参数](#)”一节第 148 页。
- **仅上载** 请参见“[Upload Only 同步参数](#)”一节第 157 页。

如果这些参数中有多个设置为 true，同步函数将抛出异常（例如，SQLCode.SQLE\_SYNC\_INFO\_INVALID 或其等价的信息）。

## Additional Parameters 同步参数

此同步参数允许应用程序提供附加参数，这些附加参数不能用任何其它的预定义参数指定。在此参数字段中，可以指定一些很少使用的参数。

这些附加参数以字符串的形式提供，字符串是由分号分隔的“关键字=值”设置值。

**语法**

根据所使用的 API 不同，语法会有所变化。

**允许值**

以下属性可以被指定为附加参数设置的一部分：

属性名称	说明
AllowDownloadDup Rows	<p>当同步遇到带重复主键的下载行时防止错误发生。</p> <p>将此属性设置为 <b>0</b> 以产生错误并回退下载；否则，设置为 <b>1</b> 以产生警告并继续下载。</p> <p>此属性仅在 UltraLite C/C++ 中可用。</p>
CheckpointStore	<p>添加数据库在同步过程中的额外检查点，以限制数据库在同步过程中的增长。</p> <p>将此属性设置为 <b>1</b> 以启用此功能，这对具有许多更新的大量下载有益，但会降低同步速度；否则，将其设置为 <b>0</b>（这是缺省值）。</p>
DisableConcurrency	<p>在上载阶段的同步期间禁止从其它线程访问数据库。</p> <p>将此属性设置为 <b>0</b> 以允许并发数据库访问；否则，将其设置为 <b>1</b>。缺省情况下，此属性被设置为 <b>0</b>。</p>
TableOrder	<p>如果 UltraLite 的缺省表排序不适合部署，则设置进行优先级同步所需的表顺序。</p> <p>将此属性设置为表名列表，表名以所需顺序排列以便上载。对于 UltraLite，使用以逗号分隔的列表；对于 ulsync，使用以分号分隔的列表。缺省情况下，顺序基于外键关系。通常，当统一数据库中的外键与 UltraLite 远程数据库匹配且不存在外键循环时，可以接受缺省值。</p> <p>使用单引号或双引号引用表名称。例如，UltraLite 支持 "Customer,Sales" 和 'Customer,Sales'。</p> <p>如果所包含的表未包括在同步中，则会忽略这些表。对没有列出的任何表均根据远程数据库中定义的外键进行适当排序。</p> <p>下载时表的顺序与为上载定义的顺序相同。</p> <p>如果 UltraLite 表出现以下情况，则只需要显式设置表顺序：</p> <ul style="list-style-type: none"> <li>● UltraLite 表是外键循环的组成部分。此时必须列出作为循环组成部分的所有表。</li> <li>● 在统一数据库中有不同的外键关系。</li> </ul>

### 另请参见

- UltraLite for C/C++: “[ul\\_synch\\_info\\_a 结构](#)”一节 《UltraLite - C 及 C++ 编程》
- UltraLite.NET: “[AdditionalParms 属性](#)”一节 《UltraLite - .NET 编程》
- UltraLite for M-Business Anywhere: “[setAdditionalParms 方法](#)”一节 《UltraLite - M-Business Anywhere 编程》

### 示例

UltraLite for C/C++ 应用程序可以按如下方法设置附加参数:

```
ul_synch_info info;
//...
info.additional_parms = UL_TEXT(
    "AllowDownloadDupRows=1;
    CheckpointStore=1;
    DisableConcurrency=1;
    TableOrder=Customer,Sales"
);
```

## Authentication Parameters 同步参数

向 MobiLink 事件中的验证参数提供参数。

### 语法

根据所使用的 API 不同语法会有所变化。

### 注释

例如, 参数可能是用户名和口令。

如果使用此参数, 还必须提供参数的数目。请参见 “[Number of Authentication Parameters 参数](#)” 一节第 145 页。

### 允许值

字符串数组。任何字符串的值都不能为 Null, 但是您可以提供一个空字符串。

### 另请参见

- “[Number of Authentication Parameters 参数](#)” 一节第 145 页
- “[验证参数](#)” 一节 《MobiLink - 服务器管理》
- “[authenticate\\_parameters 连接事件](#)” 一节 《MobiLink - 服务器管理》
- UltraLite for C/C++: “[ul\\_synch\\_info\\_a 结构](#)” 一节 《UltraLite - C 及 C++ 编程》
- UltraLite.NET: “[AuthenticationParms 属性](#)” 一节 《UltraLite - .NET 编程》
- UltraLite for M-Business Anywhere: “[setAuthenticationParms 方法](#)” 一节 《UltraLite - M-Business Anywhere 编程》

### 示例

UltraLite for C/C++ 应用程序可以按如下方法设置参数:

```
ul_char * Params[ 3 ] = { UL_TEXT( "parm1" ),
                          UL_TEXT( "parm2" ),
```

```

                                UL_TEXT( "parm3" ) };
// ...
info.num_auth_parms = 3;
info.auth_parms = Params;

```

## Authentication Status 同步参数

同步过程设置此字段来报告 MobiLink 用户验证的状态。MobiLink 服务器将此信息提供给客户端。

### 语法

根据所使用的 API 不同语法会有所变化。

### 允许值

允许值保存在接口特定的枚举中。例如，对于 C/C++ 应用程序有如下的枚举：

常量	值	说明
UL_AUTH_STATUS_UNKNOWN	0	授权状态未知，可能是因为连接尚未同步。
UL_AUTH_STATUS_VALID	1	用户 ID 和口令在同步时有效。
UL_AUTH_STATUS_VALID_BUT_EXPIRES_SOON	2	用户 ID 和口令在同步时有效，但很快就到期。
UL_AUTH_STATUS_EXPIRED	3	授权失败：用户 ID 或口令已到期。
UL_AUTH_STATUS_INVALID	4	授权失败：错误的用户 ID 或口令。
UL_AUTH_STATUS_IN_USE	5	授权失败：用户 ID 已在使用中。

### 注释

如果统一数据库上的自定义 `authenticate_user` 同步脚本返回不同的值，则根据 `authenticate_user` 连接事件中给出的规则解释该值。请参见“[authenticate\\_user 连接事件](#)”一节《[MobiLink - 服务器管理](#)》。

如果要实现自定义验证方案，则 `authenticate_user` 或 `authenticate_user_hashed` 同步脚本必须返回此参数的允许值之一。

此参数由 MobiLink 服务器设置，因此是只读参数。

### 另请参见

- “[MobiLink 用户](#)” 《[MobiLink - 客户端管理](#)》
- UltraLite for C/C++: “[ul\\_synch\\_info\\_a 结构](#)” 一节《[UltraLite - C 及 C++ 编程](#)》
- UltraLite.NET: “[AuthStatus 属性](#)” 一节《[UltraLite - .NET 编程](#)》
- UltraLite for M-Business Anywhere: “[SyncResult 类](#)” 一节《[UltraLite - M-Business Anywhere 编程](#)》

## 示例

UltraLite for C/C++ 应用程序可以按如下方法访问参数：

```
ul_synch_info info;  
//...  
returncode = info.auth_status;
```

## Authentication Value 同步参数

同步过程设置此字段来报告自定义 MobiLink 用户验证脚本的结果。MobiLink 服务器将此信息提供给向客户端。

### 语法

根据所使用的 API 不同语法会有所变化。

### 注释

在 `authenticate_user` 连接事件和 Authentication Status 同步参数中介绍了由缺省 MobiLink 用户验证机制设置的值。

此参数由 MobiLink 服务器设置，因此是只读参数。

### 另请参见

- “`authenticate_user` 连接事件” 一节 《MobiLink - 服务器管理》
- “`authenticate_user_hashed` 连接事件” 一节 《MobiLink - 服务器管理》
- “Authentication Status 同步参数” 一节第 141 页
- UltraLite for C/C++: “`ul_synch_info_a` 结构” 一节 《UltraLite - C 及 C++ 编程》
- UltraLite.NET: “AuthValue 属性” 一节 《UltraLite - .NET 编程》
- UltraLite for M-Business Anywhere: “SyncResult 类” 一节 《UltraLite - M-Business Anywhere 编程》

## 示例

UltraLite for C/C++ 应用程序可以按如下方法访问参数：

```
ul_synch_info info;  
//...  
returncode = info.auth_value;
```

## Download Only 同步参数

防止同步期间来自 UltraLite 数据库的更改被上载。

### 语法

根据所使用的 API 不同，语法会有所变化。还可以使用 `ulsync` 设置此参数。

### 缺省值

False



## 允许值

布尔值

## 与以下参数冲突

Ping 和仅上载

## 注释

当具有通过仅下载同步所同步的远程数据库时，您应该定期执行完全同步以减少仅下载同步扫描的日志量。否则，仅下载同步完成同步花费的时间将越来越长。

**对于 `ulsync`** 当发生仅下载同步时，`ulsync` 不上载任何对数据的更改。相反，它将：

- 上载有关进度计数器中存储的模式和值的信息。
- 确保在仅下载同步期间不会覆盖远程数据库上的更改。

为了实现这些操作，`ulsync` 扫描 UltraLite 数据库日志来监视统一数据库上具有待执行操作的行。如果 `ulsync` 检测到冲突，则将回退下载，而同步将失败。然后您必须执行完全同步（即上载和下载）来修正此冲突。

## 另请参见

- [“Upload Only 同步参数”一节第 157 页](#)
- [“进度计数器”一节第 116 页](#)
- [“UltraLite 同步实用程序 \(ulsync\)”一节第 255 页](#)
- UltraLite for C/C++: [“ul\\_synch\\_info\\_a 结构”一节《UltraLite - C 及 C++ 编程》](#)
- UltraLite.NET: [“DownloadOnly 属性”一节《UltraLite - .NET 编程》](#)
- UltraLite for M-Business Anywhere: [“SyncParms 类”一节《UltraLite - M-Business Anywhere 编程》](#)

## 示例

`ulsync` 支持此参数作为扩展同步参数：

```
ulsync -c DBF=myuldb.udb
"MobiLinkId=remoteA;ScriptVersion=2;DownloadOnly=ON;Stream=http"
```

UltraLite for C/C++ 应用程序可以按如下方法设置参数：

```
ul_synch_info info;
//...
info.download_only = ul_true;
```

## Ignored Rows 同步参数

同步时设置此字段来显示同步期间由于缺少脚本而被服务器忽略了的行。

## 语法

根据所使用的 API 不同语法会有所变化。

### 允许值

Boolean

### 注释

此参数是只读的。

### 另请参见

- UltraLite for C/C++: “[ul\\_synch\\_info\\_a 结构](#)”一节 《UltraLite - C 及 C++ 编程》
- UltraLite.NET: “[IgnoredRows 属性](#)”一节 《UltraLite - .NET 编程》
- UltraLite for M-Business Anywhere: “[SyncResult 类](#)”一节 《UltraLite - M-Business Anywhere 编程》

### 示例

UltraLite for C/C++ 应用程序可以按如下方法访问参数:

```
ul_synch_info info;  
//...  
res = info.ignored_rows;
```

## Keep Partial Download 同步参数

在同步期间当下载由于通信错误而失败时，此参数控制 UltraLite 是否保留部分下载而不回退更改。

### 语法

根据所使用的 API 不同语法会有所变化。

### 缺省值

False，表示 UltraLite 在下载失败后将回退所有更改。

### 允许值

布尔值

### 另请参见

- “[恢复失败的下载](#)”一节 《MobiLink - 服务器管理》
- “[Resume Partial Download 同步参数](#)”一节第 150 页
- UltraLite for C/C++: “[ul\\_synch\\_info\\_a 结构](#)”一节 《UltraLite - C 及 C++ 编程》
- UltraLite.NET: “[KeepPartialDownload 属性](#)”一节 《UltraLite - .NET 编程》
- UltraLite for M-Business Anywhere: “[SyncParms 类](#)”一节 《UltraLite - M-Business Anywhere 编程》

### 示例

UltraLite for C/C++ 应用程序可以按如下方法设置参数:

```
ul_synch_info info;  
//...  
info.keep_partial_download = ul_true;
```

## New Password 同步参数

设置与用户名关联的新 MobiLink 口令。

### 语法

根据所使用的 API 不同，语法会有所变化。还可以使用 `ulsync` 设置此参数。

### 允许值

字符串

### 注释

此参数是可选的。

### 另请参见

- “MobiLink 用户” 《MobiLink - 客户端管理》
- “UltraLite 同步实用程序 (ulsync)” 一节第 255 页
- UltraLite for C/C++: “`ul_synch_info_a` 结构” 一节 《UltraLite - C 及 C++ 编程》
- UltraLite.NET: “NewPassword 属性” 一节 《UltraLite - .NET 编程》
- UltraLite for M-Business Anywhere: “SyncParms 类” 一节 《UltraLite - M-Business Anywhere 编程》

### 示例

`ulsync` 可以将此参数按如下方法设置为扩展同步参数：

```
ulsync -c DBF=myuldb.udb
"MobiLinkUid=remoteA;ScriptVersion=2;NewMobiLinkPwd=mynewpassword;Stream=http
"
```

UltraLite for C/C++ 应用程序可以按如下方法设置参数：

```
ul_synch_info info;
// ...
info.new_password = UL_TEXT( "mlnewpass" );
```

## Number of Authentication Parameters 参数

在 MobiLink 事件中提供正在传递给验证参数的验证参数数目。

### 语法

根据所使用的 API 不同语法会有所变化。

### 缺省值

不向自定义验证脚本传递任何参数。

### 注释

此参数与验证参数一同使用，用来向自定义验证脚本提供信息。

### 另请参见

- “Authentication Parameters 同步参数” 一节第 140 页
- “authenticate\_parameters 连接事件” 一节 《MobiLink - 服务器管理》
- “验证参数” 一节 《MobiLink - 服务器管理》
- UltraLite for C/C++: “ul\_synch\_info\_a 结构” 一节 《UltraLite - C 及 C++ 编程》
- UltraLite.NET: “AuthenticationParms 属性” 一节 《UltraLite - .NET 编程》
- UltraLite for M-Business Anywhere: “SyncParms 类” 一节 《UltraLite - M-Business Anywhere 编程》

### 示例

UltraLite for C/C++ 应用程序可以按如下方法设置参数:

```
ul_synch_info info;  
//...  
info.num_auth_parms = 3;
```

## Observer 同步参数

指定一个用于监控同步的回调函数或事件处理程序的指针。

### 语法

根据所使用的 API 不同语法会有所变化。

### 另请参见

- “User Data 同步参数” 一节第 158 页
- UltraLite for C/C++: “ul\_synch\_info\_a 结构” 一节 《UltraLite - C 及 C++ 编程》
- UltraLite.NET: “ULSyncProgressListener 成员” 一节 《UltraLite - .NET 编程》
- UltraLite for M-Business Anywhere: “synchronizeWithParm 方法” 一节 《UltraLite - M-Business Anywhere 编程》

### 示例

UltraLite for C/C++ 应用程序可以按如下方法设置参数:

```
ul_synch_info info;  
//...  
info.observer=callfunction;
```

## Partial Download Retained 同步参数

在同步期间当下载由于通信错误而失败时，同步过程会设置此字段来表示 UltraLite 是否应用这些已下载的更改而不是回退这些更改。

### 语法

根据所使用的 API 不同语法会有所变化。

## 允许值

Boolean

## 注释

如果发生下载错误并且保留了部分下载，则在同步期间设置此参数。

仅当保留部分下载参数设置为 `true` 时，才会保留部分下载。请参见“[Keep Partial Download 同步参数](#)”一节第 144 页。

## 另请参见

- “恢复失败的下载”一节 《MobiLink - 服务器管理》
- “Resume Partial Download 同步参数”一节第 150 页
- UltraLite for C/C++: “ul\_synch\_info\_a 结构”一节 《UltraLite - C 及 C++ 编程》
- UltraLite.NET: “PartialDownloadRetained 属性”一节 《UltraLite - .NET 编程》
- UltraLite for M-Business Anywhere: “SyncResult 类”一节 《UltraLite - M-Business Anywhere 编程》

## 示例

按如下方法访问此参数：

```
ul_synch_info info;  
//...  
returncode=info.partial_download_retained;
```

# Password 同步参数

指定与用户名关联的 MobiLink 口令。

## 语法

根据所使用的 API 不同，语法会有所变化。还可以使用 `ulsync` 设置此参数。

## 允许值

字符串

## 注释

此参数是可选的。

此 MobiLink 用户名和口令不同于任何数据库用户 ID 和口令，仅用于向 MobiLink 服务器标识和验证应用程序。请参见“[User Name 同步参数](#)”一节第 158 页。

如果 MobiLink 客户端已有口令，则使用新口令参数来更改口令。请参见“[New Password 同步参数](#)”一节第 145 页。

### 另请参见

- “MobiLink 用户” 《MobiLink - 客户端管理》
- “UltraLite 同步实用程序 (ulsync)” 一节第 255 页
- UltraLite for C/C++: “ul\_synch\_info\_a 结构” 一节 《UltraLite - C 及 C++ 编程》
- UltraLite.NET: “Password 属性” 一节 《UltraLite - .NET 编程》
- UltraLite for M-Business Anywhere: “SyncParms 类” 一节 《UltraLite - M-Business Anywhere 编程》

### 示例

ulsync 可以将此参数按如下方法设置为扩展同步参数:

```
ulsync -c DBF=myuldb.udb  
"MobiLinkUid=remoteA;ScriptVersion=2;MobiLinkPwd=myspassword;Stream=http"
```

UltraLite for C/C++ 应用程序可以按如下方法设置参数:

```
ul_synch_info info;  
//...  
info.password = UL_TEXT( "myspassword" );
```

## Ping 同步参数

确认 UltraLite 客户端和 MobiLink 服务器之间的通信。当此参数设置为 true 时, 不进行同步。

### 语法

根据所使用的 API 不同语法会有所变化。还可以使用 ulsync 设置此参数。

### 缺省值

False

### 允许值

布尔值

### 注释

当 MobiLink 服务器接收到一个 ping 请求时, 它便会连接到统一数据库, 验证用户, 然后将验证用户状态和值发送回客户端。

如果 ping 成功, 则 MobiLink 服务器将发出一个信息消息。如果 ping 不成功, 它会发出一个错误消息。

如果在 ml\_user 系统表中找不到 MobiLink 用户 ID, 同时 MobiLink 服务器正在使用命令行选项 -zu + 运行, 则 MobiLink 服务器会将该用户添加到 ml\_user。

对于 ping 请求，MobiLink 服务器可能会执行以下脚本（如果存在）：

- begin\_connection
- authenticate\_user
- authenticate\_user\_hashed
- authenticate\_parameters
- end\_connection

### 另请参见

- “-pi 选项”一节 《MobiLink - 客户端管理》
- “UltraLite 同步实用程序 (ulsync)”一节第 255 页
- UltraLite for C/C++: “ul\_synch\_info\_a 结构”一节 《UltraLite - C 及 C++ 编程》
- UltraLite.NET: “PingOnly 属性”一节 《UltraLite - .NET 编程》
- UltraLite for M-Business Anywhere: “SyncParms 类”一节 《UltraLite - M-Business Anywhere 编程》

### 示例

ulsync 可以将此参数按如下方法设置为扩展同步参数：

```
ulsync -c DBF=myuldb.udb  
"MobiLinkUid=remoteA;ScriptVersion=2;Ping=True;Stream=http"
```

UltraLite for C/C++ 应用程序可以按如下方法设置参数：

```
ul_synch_info info;  
//...  
info.ping = ul_true;
```

## Publications 同步参数

指定要同步的发布。

### 语法

根据所使用的 API 不同语法会有所变化。还可以通过 ulsync 使用此参数。

### 缺省值

同步所有发布。

### 注释

在 C/C++ 中进行同步时，将发布同步参数设置为一个**发布列表**：以逗号分隔的发布名的列表。

### 另请参见

- “UltraLite 中的发布” 一节第 124 页
- “使用 UltraLite 发布” 一节第 75 页
- “UltraLite 同步实用程序 (ulsync)” 一节第 255 页
- UltraLite for C/C++: “ul\_synch\_info\_a 结构” 一节 《UltraLite - C 及 C++ 编程》
- UltraLite.NET: “ULPublicationSchema 类” 一节 《UltraLite - .NET 编程》
- UltraLite for M-Business Anywhere: “PublicationSchema 类” 一节 《UltraLite - M-Business Anywhere 编程》

### 示例

ulsync 可以将此参数按如下方法设置为扩展同步参数:

```
ulsync -c DBF=myuldb.udb
"MobiLinkId=remoteA;ScriptVersion=2;Publications=UL_PUB_MYPUB1,UL_PUB_MYPUB2
;Stream=http"
```

UltraLite for C/C++ 应用程序可以按如下方法设置参数:

```
ul_synch_info info;
//...
info.publications = UL_TEXT( "Pubs1,Pubs3" );
```

## Resume Partial Download 同步参数

重新开始失败的下载。

### 语法

根据所使用的 API 不同语法会有所变化。还可以使用 ulsync 设置此参数。

### 缺省值

False

### 允许值

布尔值

### 注释

同步不上载更改; 它仅下载那些要在失败的下载中下载的更改。



## 另请参见

- “恢复失败的下载”一节 《MobiLink - 服务器管理》
- “Keep Partial Download 同步参数”一节第 144 页
- “Partial Download Retained 同步参数”一节第 146 页
- “UltraLite 同步实用程序 (ulsync)”一节第 255 页
- UltraLite for C/C++: “ul\_synch\_info\_a 结构”一节 《UltraLite - C 及 C++ 编程》
- UltraLite.NET: “ResumePartialDownload 属性”一节 《UltraLite - .NET 编程》
- UltraLite for M-Business Anywhere: “SyncParms 类”一节 《UltraLite - M-Business Anywhere 编程》

## 示例

UltraLite for C/C++ 应用程序可以按如下方法设置参数:

```
ul_synch_info info;
//...
info.resume_partial_download = ul_true;
```

## Send Column Names 同步参数

指定上载中应该发送列名。

## 语法

根据所使用的 API 不同语法会有所变化。还可以使用 `ulsync` 设置此参数。

## 缺省值

False

## 允许值

布尔值

## 注释

此选项由 MobiLink 服务器用于直接行处理。使用直接行处理时，应启用此选项。否则，它将不起作用。请参见“直接行处理” 《MobiLink - 服务器管理》。

## 另请参见

- “UltraLite 同步实用程序 (ulsync)”一节第 255 页
- UltraLite for C/C++: “ul\_synch\_info\_a 结构”一节 《UltraLite - C 及 C++ 编程》
- UltraLite.NET: “SendColumnNames 属性”一节 《UltraLite - .NET 编程》
- UltraLite for M-Business Anywhere: “SyncParms 类”一节 《UltraLite - M-Business Anywhere 编程》

## 示例

`ulsync` 可以将此参数按如下方法设置为扩展同步参数:

```
ulsync -c DBF=myuldb.udb
"MobiLinkUid=remoteA;ScriptVersion=2;SendColumnNames=true;Stream=http"
```

UltraLite for C/C++ 应用程序可以按如下方法设置参数:

```
ul_synch_info info;  
//...  
info.send_column_names = ul_true;
```

## 发送下载确认同步参数

指示 MobiLink 服务器此客户端将提供下载确认。

### 语法

根据所使用的 API 不同语法会有所变化。还可以使用 `ulsync` 设置此参数。

### 缺省值

False

### 允许值

布尔值

### 另请参见

- [“UltraLite 同步实用程序 \(ulsync\)” 一节第 255 页](#)
- UltraLite for C/C++: [“ul\\_synch\\_info\\_a 结构” 一节 《UltraLite - C 及 C++ 编程》](#)
- UltraLite.NET: [“SendDownloadAck 属性” 一节 《UltraLite - .NET 编程》](#)
- UltraLite for M-Business Anywhere: [“SyncParms 类” 一节 《UltraLite - M-Business Anywhere 编程》](#)

### 示例

`ulsync` 可以将此参数按如下方法设置为扩展同步参数:

```
ulsync -c DBF=myuldb.udb  
"MobiLinkUid=remoteA;ScriptVersion=2;SendDownloadACK=true;Stream=http"
```

UltraLite for C/C++ 应用程序可以按如下方法设置参数:

```
ul_synch_info info;  
//...  
info.send_download_ack = ul_true;
```

## Stream Error 同步参数

提供一个用来保存通信错误报告信息的结构。

### 语法

根据所使用的 API 不同, 语法会有所变化。

## 适用于

此参数仅适用于 C/C++ 接口。

## 允许值

此参数没有缺省值，因而必须使用支持的字段之一进行显式设置。ul\_stream\_error 字段如下所示：

- **stream\_error\_code** 非必需字段。值始终为 0。  
有关错误号的列表，请参见“[MobiLink 通信错误消息](#)”《[错误消息](#)》。有关错误代码后缀的信息，请参见 *install-dir\SDK\Include\sserror.h*。
- **system\_error\_code** 系统特定的错误代码。有关错误代码的详细信息，必须查看平台文档。对于 Windows 平台，此文档是 Microsoft Developer Network 文档。  
以下是 Windows 上的常见系统错误：
  - **10048 (WSAADDRINUSE)** 地址已经在使用。
  - **10053 (WSAECONNABORTED)** 软件导致连接中止。
  - **10054 (WSAECONNRESET)** 通信的另一端关闭了套接字。
  - **10060 (WSAETIMEDOUT)** 连接超时。
  - **10061 (WSAECONNREFUSED)** 连接被拒绝。通常情况下，这表示 MobiLink 服务器没有运行或者没有在指定的端口上监听。请参见 [Microsoft Developer Network Web 站点](#)。
- **error\_string** 应用程序提供的错误消息。该字符串可能是空的，也可能不是空的。除 stream\_error\_code 以外，非空的 error\_string 还提供其它信息。例如，对于写入错误（错误代码 9），错误字符串是一个表示试图写入的字节数的数字。
- **error\_string\_length** 不建议使用。错误字符串缓冲区的大小。

## 注释

UltraLite 应用程序（UltraLite C++ 组件除外）将通信错误信息作为同步结果参数的一部分接收。请参见“[Sync Result 同步参数](#)”一节第 156 页。

stream\_error 字段是类型为 ul\_stream\_error 的结构。

```
typedef struct {
    ss_error_code stream_error_code;
    asa_uint16 alignment;
    asa_int32 system_error_code;
    char error_string[UL_STREAM_ERROR_STRING_SIZE];
} ul_stream_error, * p_ul_stream_error;
```

结构被定义在 *install-dir\SDK\Include\sserror.h* 文件中。

检查 SQLE\_COMMUNICATIONS\_ERROR:

```
Connection conn;
ul_synch_info info;
...
conn.InitSynchInfo( &info );
info.stream_error.error_string = error_buff;
info.stream_error.error_string_length =
    sizeof( error_buff );
```

```

if( !conn.Synchronize( &synch_info ) ){
    if( SQLCODE == SQLCOMMUNICATIONS_ERROR ){
        printf( error_buff );
        // more error handline here
    }
}

```

**另请参见**

- “ul\_synch\_info\_a 结构”一节 《UltraLite - C 及 C++ 编程》

## Stream Type 同步参数

设置用于同步的 MobiLink 网络协议。

**语法**

根据所使用的 API 不同，语法会有所变化。还可以使用 `ulsync` 设置此参数。

**注释**

该参数是必需的。它没有缺省值。

大多数网络协议要求协议选项标识 MobiLink 服务器地址和其它行为。这些选项在 Stream Parameters 参数中提供。请参见“Stream Parameters 同步参数”一节第 155 页。

如果网络协议需要选项，可使用 Stream Parameters 参数传递该选项；否则，将 Stream Parameters 参数设置为空。

以下流类型可用，但并非所有类型在所有目标平台上都可用：

网络协议	说明
HTTP	通过 HTTP 进行同步。
HTTPS	通过 HTTPS 进行同步。 HTTPS 协议使用 TLS 作为其基础安全层。它通过 TCP/IP 进行操作。
TCP/IP	通过 TCP/IP 进行同步。
TLS	使用传送层安全（Transport-Layer Security，简称 TLS）通过 TCP/IP 进行同步。TLS 使用数字证书和公共密钥加密算法保护客户端/服务器的通信安全。

有关受支持的平台的列表，请参见 <http://www.sybase.com/detail?id=1062623>。

## 另请参见

- “证书创建实用程序 (createcert)” 一节 《SQL Anywhere 服务器 - 数据库管理》
- “证书查看器实用程序 (viewcert)” 一节 《SQL Anywhere 服务器 - 数据库管理》
- “传送层安全” 《SQL Anywhere 服务器 - 数据库管理》
- “UltraLite 同步实用程序 (ulsync)” 一节第 255 页
- “UltraLite 同步流的网络协议选项” 一节第 161 页
- UltraLite for C/C++: “ul\_synch\_info\_a 结构” 一节 《UltraLite - C 及 C++ 编程》
- UltraLite.NET: “Stream 属性” 一节 《UltraLite - .NET 编程》
- UltraLite for M-Business Anywhere: “SyncParms 类” 一节 《UltraLite - M-Business Anywhere 编程》

## 示例

对于 UltraLite for C/C++ 应用程序，按如下方法设置参数：

```
Connection conn;
ul_synch_info info;
...
conn.InitSynchInfo( &info );
info.stream = "http";
```

## Stream Parameters 同步参数

设置选项来配置网络协议。

## 语法

根据所使用的 API 不同语法会有所变化。还可以使用 ulsync 设置此参数。

## 缺省值

Null

## 允许值

字符串

## 注释

此参数是可选的。它接受用分号分隔的网络协议选项列表。每个选项的形式均为 *keyword=value*，其中允许使用的关键字集取决于网络协议。

## 另请参见

- “UltraLite 同步实用程序 (ulsync)” 一节第 255 页
- “UltraLite 同步流的网络协议选项” 一节第 161 页
- UltraLite for C/C++: “ul\_synch\_info\_a 结构” 一节 《UltraLite - C 及 C++ 编程》
- UltraLite.NET: “StreamParms 属性” 一节 《UltraLite - .NET 编程》
- UltraLite for M-Business Anywhere: “SyncParms 类” 一节 《UltraLite - M-Business Anywhere 编程》

## 示例

UltraLite for C/C++ 应用程序可以按如下方法设置参数：

```
ul_synch_info info;  
//...  
info.stream_parms= UL_TEXT( "host=myserver;port=2439" );
```

## Sync Result 同步参数

报告同步的状态。

### 语法

根据所使用的 API 不同，语法会有所变化。

### 注释

此参数由 UltraLite 设置，并且是只读参数。

C/C++ 接口以单独的参数形式将此信息作为 `ul_synch_info` 结构的一部分接收。否则，将此信息定义为一个复合参数，包含多种信息，分别使用不同的字段：

- **Authentication Status** 报告验证是成功还是失败。请参见 [“Authentication Status 同步参数”一节第 141 页](#)。
- **Ignored Rows** 报告忽略的行的数目。请参见 [“Ignored Rows 同步参数”一节第 143 页](#)。
- **Stream Error information** 流错误信息包括流错误代码、流错误上下文、流错误 ID 和流错误系统。请参见 [“Stream Error 同步参数”一节第 152 页](#)。
- **Upload OK** 报告上载阶段是成功还是失败。请参见 [“Upload OK 同步参数”一节第 156 页](#)。

### 另请参见

- UltraLite.NET: [“ULSyncParms 类”一节《UltraLite - .NET 编程》](#)
- UltraLite C/C++: [“ul\\_synch\\_result 结构”一节《UltraLite - C 及 C++ 编程》](#)
- UltraLite for M-Business Anywhere: [“SyncParms 类”一节《UltraLite - M-Business Anywhere 编程》](#)
- UltraLite for embedded SQL: [“ULGetSynchResult 函数”一节《UltraLite - C 及 C++ 编程》](#)

## Upload OK 同步参数

同步过程设置此字段来报告上载到 MobiLink 服务器的数据的状态。

### 语法

根据所使用的 API 不同，语法会有所变化。

### 注释

此参数由 UltraLite 设置，因此是只读参数。

同步后，如果上载成功，该参数保持为 **true**，否则为 **false**。如果有同步错误，您可以检查此参数，以了解在错误发生以前数据是否已经成功上载。

### 另请参见

- UltraLite C/C++: “[ul\\_synch\\_info\\_a 结构](#)”一节 《UltraLite - C 及 C++ 编程》
- UltraLite.NET: “[UploadOK 属性](#)”一节 《UltraLite - .NET 编程》
- UltraLite for M-Business Anywhere: “[SyncResult 类](#)”一节 《UltraLite - M-Business Anywhere 编程》

### 示例

UltraLite for C/C++ 应用程序可以按如下方法访问参数:

```
ul_synch_info info;  
//...  
returncode = info.upload_ok;
```

## Upload Only 同步参数

表示在当前同步过程中不应进行任何下载，这样可以节省通信时间，尤其是在通过慢速通信链接进行通信时。

### 语法

根据所使用的 API 不同语法会有所变化。还可以使用 `ulsync` 设置此参数。

### 缺省值

False

### 允许值

布尔值

### 与以下参数冲突

仅下载、Ping 和恢复部分下载

### 注释

当设置为 **true** 时，客户端将等待来自 MobiLink 服务器的上载确认，得到确认之后它就会成功地终止同步会话。

### 另请参见

- “[设计 UltraLite 中的同步](#)”一节第 123 页
- “[Download Only 同步参数](#)”一节第 142 页
- “[UltraLite 同步实用程序 \(ulsync\)](#)”一节第 255 页
- UltraLite C/C++: “[ul\\_synch\\_info\\_a 结构](#)”一节 《UltraLite - C 及 C++ 编程》
- UltraLite.NET: “[UploadOnly 属性](#)”一节 《UltraLite - .NET 编程》
- UltraLite for M-Business Anywhere: “[SyncParms 类](#)”一节 《UltraLite - M-Business Anywhere 编程》

## 示例

ulsync 可以将此参数按如下方法设置为扩展同步参数：

```
ulsync -c DBF=myuldb.udb  
"MobiLinkUid=remoteA;ScriptVersion=2;UploadOnly=True;Stream=http"
```

UltraLite for C/C++ 应用程序可以按如下方法设置参数：

```
ul_synch_info info;  
//...  
info.upload_only = ul_true;
```

## User Data 同步参数

使应用程序特定的信息可用于同步观察器。

### 适用于

仅适用于 C/C++ 应用程序。其它组件（如 UltraLite.NET）并不需要单独的参数来处理用户数据，因而没有 User Data 参数。

### 语法

根据所使用的 API 不同，语法会有所变化。

### 注释

实现同步观察器回调函数或事件处理程序时，使用应用程序专用的信息的方法是使用 User Data 参数提供信息。

### 另请参见

- “Observer 同步参数”一节第 146 页
- UltraLite C/C++：“ul\_synch\_info\_a 结构”一节 《UltraLite - C 及 C++ 编程》

## User Name 同步参数

必需。MobiLink 服务器用于验证目的的字符串。

### 语法

根据所使用的 API 不同，语法会有所变化。还可以使用 ulsync 设置此参数。

### 注释

该参数是必需的。通常会拒绝空字符串和 NULL 字符串。

此参数没有缺省值，必须显式设置。

使用远程 ID 时，用户名不必唯一。请参见“远程 ID”一节 《MobiLink - 客户端管理》。

此 MobiLink 用户名和口令不同于任何数据库用户 ID 和口令，仅用于向 MobiLink 服务器识别和验证应用程序。请参见“Password 同步参数”一节第 147 页。



对于作为同步系统的组成部分的用户，您必须使用 MobiLink 服务器注册用户名。此用户名存储在统一数据库的 ml\_user MobiLink 系统表的名称列中。

### 另请参见

- “MobiLink 用户” 《MobiLink - 客户端管理》
- “UltraLite 用户验证” 一节 《MobiLink - 客户端管理》
- “UltraLite 同步实用程序 (ulsync)” 一节第 255 页
- UltraLite C/C++: “ul\_synch\_info\_a 结构” 一节 《UltraLite - C 及 C++ 编程》
- UltraLite.NET: “UserName 属性” 一节 《UltraLite - .NET 编程》
- UltraLite for M-Business Anywhere: “SyncParms 类” 一节 《UltraLite - M-Business Anywhere 编程》

### 示例

ulsync 可以将此参数按如下方法设置为扩展同步参数:

```
ulsync -c DBF=myuldb.udb "MobiLinkUid=remoteA;ScriptVersion=2;Stream=http"
```

UltraLite for C/C++ 应用程序可以按如下方法设置参数:

```
ul_synch_info info;
//...
info.user_name= UL_TEXT( "remoteA" );
```

## Version 同步参数

定义统一数据库版本。

### 语法

根据所使用的 API 不同，语法会有所变化。还可以使用 ulsync 设置此参数。

### 允许值

字符串

### 注释

该参数是必需的。通常会拒绝空字符串和 NULL 字符串。

统一数据库中的每个同步脚本都带有版本字符串标记。例如，可能有两个不同的 download\_cursor 脚本，分别由不同的版本字符串标识。

### 另请参见

- “脚本版本” 一节 《MobiLink - 服务器管理》
- “UltraLite 同步实用程序 (ulsync)” 一节第 255 页
- UltraLite C/C++: “ul\_synch\_info\_a 结构” 一节 《UltraLite - C 及 C++ 编程》
- UltraLite.NET: “Version 属性” 一节 《UltraLite - .NET 编程》
- UltraLite for M-Business Anywhere: “SyncParms 类” 一节 《UltraLite - M-Business Anywhere 编程》

### 示例

ulsync 可以将此参数按如下方法设置为扩展同步参数:

```
ulsync -c DBF=myuldb.udb "MobiLinkId=remoteA;ScriptVersion=2;Stream=http"
```

UltraLite for C/C++ 应用程序可以按如下方法设置参数:

```
ul_synch_info info;  
//...  
info.version = UL_TEXT( "default" );
```

## UltraLite 同步流的网络协议选项

必须在应用程序中设置网络协议。每个与 MobiLink 服务器同步的 UltraLite 数据库均通过网络协议进行同步。可用的网络协议包括 TCP/IP、HTTP、HTTPS 和 TLS。还对 Palm OS 上的 HotSync 同步及 Windows CE 上的 ActiveSync 通知提供支持。

对于所设置的网络协议，可以从一组相应的协议选项中进行选择，以确保 UltraLite 应用程序可以找到 MobiLink 服务器并与其正确地通信。网络协议选项提供诸如寻址信息（主机和端口）和协议特定信息之类的信息。参考下表确定可将哪些选项用于您使用的流类型。

有关协议选项的列表，请参见“[MobiLink 客户端网络协议选项汇总](#)”一节《[MobiLink - 客户端管理](#)》。

### 另请参见

- “[配置 UltraLite 客户端以使用传送层安全](#)”一节《[SQL Anywhere 服务器 - 数据库管理](#)》
- “[通过启用 TLS 的同步部署 UltraLite](#)”一节第 51 页
- “[MobiLink 客户端网络协议选项](#)”《[MobiLink - 客户端管理](#)》
- “[Stream Parameters 同步参数](#)”一节第 155 页
- “[UltraLite 同步实用程序 \(ulsync\)](#)”一节第 255 页中的 -x 选项

## 设置同步流和选项

可以通过设置“流参数”参数来提供在应用程序中查找 MobiLink 服务器所需的信息。请参见“[Stream Parameters 同步参数](#)”一节第 155 页。

不过，如果您正在使用 HotSync，并且没有指定流参数值，或者即使已将流参数值指定为 NULL，您也可以从 HotSync 管理器中输入所需参数。请参见“[设置 MobiLink 同步的协议选项](#)”一节第 133 页。

有关在 UltraLite 同步调用中包括流参数的信息，请参见：

- UltraLite for C/C++： “[将 HotSync 同步添加到 Palm 应用程序](#)”一节《[UltraLite - C 及 C++ 编程](#)》
- UltraLite for UltraLite.NET： “[StreamParms 属性](#)”一节《[UltraLite - .NET 编程](#)》
- UltraLite for M-Business Anywhere： “[setStreamParms 方法](#)”一节《[UltraLite - M-Business Anywhere 编程](#)》

### **zlib 压缩注释**

在 Palm OS 上不支持 zlib 压缩。

---

# UltraLite 数据库参考

本节提供 UltraLite 数据库属性、选项、连接参数和实用程序的参考资料。

---

UltraLite 创建参数 .....	165
UltraLite 数据库属性 .....	195
UltraLite 数据库选项 .....	201
UltraLite 连接参数 .....	207
UltraLite 实用程序 .....	227
UltraLite 系统表 .....	267



---

# UltraLite 创建参数

## 目录

UltraLite 大小写创建参数 .....	167
UltraLite checksum_level 创建参数 .....	168
UltraLite 归类创建参数 .....	170
UltraLite date_format 创建参数 .....	171
UltraLite date_order 创建参数 .....	174
UltraLite fips 创建参数 .....	176
UltraLite max_hash_size 创建参数 .....	177
UltraLite nearest_century 创建参数 .....	179
UltraLite obfuscate 创建参数 .....	181
UltraLite page_size 创建参数 .....	182
UltraLite 精度创建参数 .....	184
UltraLite 小数位数创建参数 .....	185
UltraLite time_format 创建参数 .....	187
UltraLite timestamp_format 创建参数 .....	189
UltraLite timestamp_increment 创建参数 .....	192
UltraLite utf8_encoding 创建参数 .....	193

---

创建参数可用于首次创建 UltraLite 数据库时对其进行配置。只能通过重新创建数据库来更改这些设置。

可以在使用 `ulcreate`、`ulinit` 或 `ulload` 实用程序以及从受支持的客户端应用程序中创建数据库时指定创建参数。

布尔创建参数用 YES、Y、ON、TRUE、T 或 1 打开，用 NO、N、OFF、FALSE、F 和 0 之一关闭。这些参数不区分大小写。

UltraLite 创建参数用以分号分隔的字符串来指定。例如：

```
ulcreate -o "case=respect;utf_encoding=1" -y test.udb
```

或者，可以指定多个 `-o` 选项。

**另请参见**

- [“UltraLite 创建数据库实用程序 \(ulcreate\)” 一节第 235 页](#)
- [“UltraLite 初始化数据库实用程序 \(ulinit\)” 一节第 249 页](#)
- [“UltraLite 装载 XML 到数据库实用程序 \(ulload\)” 一节第 252 页](#)
- [“访问创建参数值” 一节第 30 页](#)



## UltraLite 大小写创建参数

将 UltraLite 数据库中的字符串比较设置成区分大小写。

### 语法

```
{ ulcreate | ulinit | ulload } -o case=value;...
```

### 允许值

Ignore、Respect

### 缺省值

Ignore

### 注释

数据的区分大小写会反映在表、索引等内容中。缺省情况下，虽然数据始终保持输入时的大小写，但 UltraLite 数据库在比较时不区分大小写。标识符（如表名和列名）和用户 ID 始终区分大小写，不管数据库是否区分大小写。无论数据库是否区分大小写，口令始终是区分大小写的。请参见“[UltraLite 中的字符串](#)”一节第 282 页。

字符串的比较结果和排序顺序部分取决于数据库是否区分大小写。

在假定标识符不区分大小写时，对于某些归类需要特别注意。特别是，土耳其语归类的大小写转换行为会导致意想不到且难以察觉的错误。最常见的错误是找不到包含字母 i 或 I 的系统对象。

您不能更改现有数据库的大小写。而是必须创建一个新数据库。

从 Sybase Central 中，可以在任何一个创建数据库向导中设置区分大小写。在 [[新数据库归类和字符集](#)] 页面上，选择 [[使用区分大小写字符串比较](#)] 选项。

在客户端应用程序中，对于数据库管理器类，将此参数设置为创建数据库方法的创建参数之一。

### 另请参见

- “[UltraLite 创建数据库实用程序 \(ulcreate\)](#)” 一节第 235 页
- “[UltraLite 初始化数据库实用程序 \(ulinit\)](#)” 一节第 249 页
- “[UltraLite 装载 XML 到数据库实用程序 \(ulload\)](#)” 一节第 252 页
- UltraLite for embedded SQL: “[ULCreateDatabase 函数](#)” 一节 《[UltraLite - C 及 C++ 编程](#)》
- UltraLite for C++: “[CreateDatabase 函数](#)” 一节 《[UltraLite - C 及 C++ 编程](#)》
- UltraLite.NET: “[CreateDatabase 方法](#)” 一节 《[UltraLite - .NET 编程](#)》
- UltraLite for M-Business Anywhere: “[createDatabase 方法](#)” 一节 《[UltraLite - M-Business Anywhere 编程](#)》
- “[访问创建参数值](#)” 一节第 30 页

## UltraLite checksum\_level 创建参数

设置数据库中校验和校验的级别。

### 语法

```
{ ulcreate | ulinit | ulload } -o checksum_level=value;...
```

### 允许值

0, 1, 2

### 缺省值

0

### 注释

校验和用于检测存储到磁盘、闪存或内存的页上的脱机损坏情况，这有助于降低由于重要页面发生错误而损坏其它数据的可能性。视所选择的级别而定，UltraLite 在将数据库页写入存储器之前计算和记录每页的校验和。

如果从存储器中读取某页时，计算出的校验和与存储的校验和不符，则表明该页在页的存储/检索过程中已被修改或已损坏。如果校验和校验失败，则在数据库载入页面时，UltraLite 将停止数据库并报告致命错误。此错误无法更正；您必须重新创建 UltraLite 数据库并向 iAnywhere 报告数据库故障。

如果在卸载和重装 UltraLite 数据库时启用了校验和，将保留并恢复校验和级别。

checksum\_level 支持以下值：

- **0** 不向数据库页添加校验和。
- **1** 向重要的数据库页（如索引和同步状态页）添加校验和，但不为行页添加校验和。
- **2** 向所有数据库页添加校验和。

从 Sybase Central 中，可以在任何一个创建数据库向导中设置校验和的使用。在 [\[新数据库存储设置\]](#) 页面上，选择 [\[数据库页的校验和级别\]](#) 选项。

在客户端应用程序中，对于数据库管理器类，将此参数设置为创建数据库方法的创建参数之一。

**另请参见**

- checksum\_level 数据库属性: “UltraLite 数据库属性” 第 195 页
- “UltraLite 创建数据库实用程序 (ulcreate)” 一节第 235 页
- “UltraLite 初始化数据库实用程序 (ulinit)” 一节第 249 页
- “UltraLite 装载 XML 到数据库实用程序 (ulload)” 一节第 252 页
- UltraLite for embedded SQL: “ULCreateDatabase 函数” 一节 《UltraLite - C 及 C++ 编程》
- UltraLite for C++: “CreateDatabase 函数” 一节 《UltraLite - C 及 C++ 编程》
- UltraLite.NET: “CreateDatabase 方法” 一节 《UltraLite - .NET 编程》
- UltraLite for M-Business Anywhere: “createDatabase 方法” 一节 《UltraLite - M-Business Anywhere 编程》
- “UltraLite 优化策略” 一节第 112 页
- “UltraLite 性能与优化” 第 99 页
- “UltraLite page\_size 创建参数” 一节第 182 页
- “连接到 UltraLite 数据库” 第 37 页

## UltraLite 归类创建参数

设置数据库归类。

### 语法

```
{ ulcreate | ulinit | ulload } -o collation=value;...
```

### 允许值

字符串

### 缺省值

1252Latin1

### 注释

有关 UltraLite 支持的归类列表，请参见“UltraLite 支持的归类”一节第 33 页。

还可以通过执行以下命令查看 UltraLite 支持的归类列表：

```
ulcreate -l
```

从 Sybase Central 中，可以在任何一个创建数据库向导中设置归类。在 [新数据库归类和字符集] 页面上，选择缺省归类 (1252Latin1) 或者从列表中选择另外一个归类。

在客户端应用程序中，对于数据库管理器类，将此参数设置为创建数据库方法的创建参数之一。

### 另请参见

- “UltraLite 创建数据库实用程序 (ulcreate)” 一节第 235 页
- “UltraLite 初始化数据库实用程序 (ulinit)” 一节第 249 页
- “UltraLite 装载 XML 到数据库实用程序 (ulload)” 一节第 252 页
- “UltraLite 字符集” 一节第 32 页
- UltraLite for embedded SQL: “ULCreateDatabase 函数” 一节 《UltraLite - C 及 C++ 编程》
- UltraLite for C++: “CreateDatabase 函数” 一节 《UltraLite - C 及 C++ 编程》
- UltraLite.NET: “CreateDatabase 方法” 一节 《UltraLite - .NET 编程》
- UltraLite for M-Business Anywhere: “createDatabase 方法” 一节 《UltraLite - M-Business Anywhere 编程》
- “访问创建参数值” 一节第 30 页

## UltraLite date\_format 创建参数

为从数据库检索得来的日期设置格式。

### 语法

```
{ ulcreate | ulinit | ulload } -o date_format=value;...
```

### 允许值

String

### 缺省值

YYYY-MM-DD（这种格式符合 ISO 日期格式规范）

### 注释

DATE 数据类型值以 date\_format 创建参数所设置的格式表示。不过，也可以用字符串表示日期值。在该值可以被检索之前，必须将其指派给一个字符串。

UltraLite 从日期部分构建日期。日期部分可包括年、月、月中第几天、周中第几天、年中第几天、小时、分钟、秒（及其小数部分）。

ISO (YYYY-MM-DD) 是缺省的日期格式和顺序。例如，以此国际标准格式表示的 "2006 年 1 月 7 日" 会写为：2006-01-07。如果您不想使用缺省 ISO 日期格式和顺序，必须为这些日期部分指定另一不同的格式和顺序。

格式为使用以下符号的字符串：

符号	说明
<i>yy</i>	两位数表示的年份。
<i>yyyy</i>	四位数表示的年份。
<i>mm</i>	两位数表示的月份，如果跟在冒号后面（例如， <i>hh:mm</i> ），则为两位数表示的分钟数。
<i>mmm</i> [ <i>m...</i> ]	月份的字符简写形式—有多少个 "m"，就有多少个字符。大写字母 M 会使输出也是大写字母。
<i>d</i>	一位数表示的周中的某一天（0 代表星期日，6 代表星期六）。
<i>dd</i>	两位数表示的月中的某一天。不需要前导零。
<i>ddd</i> [ <i>d...</i> ]	周中某天的字符简写形式。大写字母 D 会使输出也是大写字母。
<i>hh</i>	两位数表示的小时数。不需要前导零。
<i>nn</i>	两位数表示的分钟数。不需要前导零。

符号	说明
<i>ss</i> [ <i>.ss..</i> ]	秒及秒的小数部分。
<i>aa</i>	使用 12 小时制。用 AM 表示上午的时间。
<i>pp</i>	使用 12 小时制。用 PM 表示下午的时间。
<i>jjj</i>	年中的某一天，从 1 到 366。

您不能更改现有数据库的日期格式。而是必须创建一个新数据库。

允许的值是用上表中列出的符号构造的。每个符号都会替换为按此设置格式的相应日期值。

对于字符简写形式，计算指定的字母数。如有必要，还可以将 A.M. 或 P.M. 指示符（可以本地化）截断为与指定的字符数相应的字节数。

**控制输出大小写** 对于表示字符数据的符号（例如 *mmm*），可以控制输出的大小写，如下所示：

- 以大写形式输入符号，使格式以大写形式显示。例如，MMM 会生成 JAN 形式的输出。
- 以小写形式输入符号，使格式以小写形式显示。例如，mmm 会生成 jan 形式的输出。
- 以大小写混合的形式输入符号，让 UltraLite 针对所用语言来选择适当的大小写。例如，以英语输入 Mmm 会生成 May 形式的输出，而在法语中会生成 mai 形式的输出。

#### 控制零填充

对于表示数字数据的符号，您可以根据符号的大小写情况来控制是否填充零：

- 以同样的大小写形式（例如 MM 或 mm）输入符号，可允许以零填充。例如，yyyy/mm/dd 可生成 2002/01/01 形式的输出。
- 以大小写混合的形式（例如 Mm）输入字符会取消以零填充。例如，yyyy/Mm/Dd 可生成 2002/1/1 形式的输出。

从 Sybase Central 中，可以在任何一个创建数据库向导中设置日期格式。在 [[新数据库创建参数](#)] 页面上，选择 [[日期格式](#)] 选项。

在客户端应用程序中，对于数据库管理器类，将此参数设置为创建数据库方法的创建参数之一。

#### 另请参见

- [“UltraLite date\\_order 创建参数”一节第 174 页](#)
- [“UltraLite 创建数据库实用程序 \(ulcreate\)”一节第 235 页](#)
- [“UltraLite 初始化数据库实用程序 \(ulinit\)”一节第 249 页](#)
- [“UltraLite 装载 XML 到数据库实用程序 \(ulload\)”一节第 252 页](#)
- UltraLite for embedded SQL: [“ULCreateDatabase 函数”一节《UltraLite - C 及 C++ 编程》](#)
- UltraLite for C++: [“CreateDatabase 函数”一节《UltraLite - C 及 C++ 编程》](#)
- UltraLite.NET: [“CreateDatabase 方法”一节《UltraLite - .NET 编程》](#)
- UltraLite for M-Business Anywhere: [“createDatabase 方法”一节《UltraLite - M-Business Anywhere 编程》](#)
- [“访问创建参数值”一节第 30 页](#)

**示例**

下表介绍了 date\_format 设置与来自 SELECT CURRENT DATE 语句的输出，执行时间为 2001 年 5 月 21 日，星期四。

使用的 date_format 语法	返回的结果
<i>yyyy/mm/ddddd</i>	2001/05/21/thu
<i>jji</i>	141
<i>mmm yyyy</i>	may 2001
<i>mm-yyyy</i>	05-2001

## UltraLite date\_order 创建参数

控制日期格式的解释。

### 语法

```
{ ulcreate | ulinit | ulload } -o date_order=value;...
```

### 允许值

MDY、YMD、DMY

### 缺省值

YMD（此格式符合 ISO 日期格式规范）

### 注释

DATE 数据类型值以 date\_format 创建参数所设置的格式表示。不过，也可以用字符串表示日期值。在该值可以被检索之前，必须将其指派给一个字符串。

UltraLite 从日期部分构建日期。日期部分可包括年、月、月中第几天、周中第几天、年中第几天、小时、分钟、秒（及其小数部分）。

ISO (YYYY-MM-DD) 是缺省的日期格式和顺序。例如，以此国际标准格式表示的 "2006 年 1 月 7 日" 会写为：2006-01-07。如果您不想使用缺省 ISO 日期格式和顺序，必须为这些日期部分指定另一不同的格式和顺序。

您不能更改现有数据库的日期顺序。而是必须创建一个新数据库。

从 Sybase Central 中，可以在任何一个创建数据库向导中设置日期顺序。在 [[新数据库创建参数](#)] 页面上，选择 [[日期顺序](#)] 选项。

在客户端应用程序中，对于数据库管理器类，将此参数设置为创建数据库方法的创建参数之一。

### 另请参见

- [“UltraLite date\\_format 创建参数”一节第 171 页](#)
- [“UltraLite 创建数据库实用程序 \(ulcreate\)”一节第 235 页](#)
- [“UltraLite 初始化数据库实用程序 \(ulinit\)”一节第 249 页](#)
- [“UltraLite 装载 XML 到数据库实用程序 \(ulload\)”一节第 252 页](#)
- UltraLite for embedded SQL: [“ULCreateDatabase 函数”一节《UltraLite - C 及 C++ 编程》](#)
- UltraLite for C++: [“CreateDatabase 函数”一节《UltraLite - C 及 C++ 编程》](#)
- UltraLite.NET: [“CreateDatabase 方法”一节《UltraLite - .NET 编程》](#)
- UltraLite for M-Business Anywhere: [“createDatabase 方法”一节《UltraLite - M-Business Anywhere 编程》](#)
- [“访问创建参数值”一节第 30 页](#)

### 示例

不同的值决定着 10/11/12 这个日期的转换形式：



---

使用的语法	转换
MDY	Oct 11 1912
YMD	Nov 12 1910
DMY	Nov 10 1912

## UltraLite fips 创建参数

控制新数据库应该使用 AES 还是 AES\_FIPS 高度加密来进行加密。

### 语法

```
{ ulcreate | ulinit | ulload } -o fips=value;KEY=value;...
```

### 允许值

是（使用 AES\_FIPS），否（使用 AES）

### 缺省值

是

### 注释

更改数据库加密类型的唯一方法是通过恰当的 fips 或 obfuscate 创建参数来重新创建数据库。可以通过在 Connection 对象上指定一个新加密密钥来更改数据库加密密钥。连接到数据库的用户必须在他们每次连接时提供密钥。

从 Sybase Central 中，可以在任何一个创建数据库向导中配置高度加密。在 [\[新数据库存储设置选项\]](#) 页面上，选择 [\[AES FIPS 算法\]](#) 选项。还必须设置和确认加密密钥。

在 Palm OS 上，使用 ULEnableFipsStrongEncryption 方法。

在所有其它平台上，对于数据库管理器类，将此参数设置为创建数据库方法的创建参数之一。

要部署启用了 FIPS 的数据库，针对您的平台复制所有适当的库。请参见 [“通过 AES\\_FIPS 数据库加密部署 UltraLite”](#) 一节第 50 页。

### 另请参见

- [“高度加密”](#) 一节 《SQL Anywhere 服务器 - 数据库管理》
- [“保护 UltraLite 数据库”](#) 一节第 35 页
- [“UltraLite obfuscate 创建参数”](#) 一节第 181 页
- [“UltraLite DBKEY 连接参数”](#) 一节第 214 页
- [“UltraLite 创建数据库实用程序 \(ulcreate\)”](#) 一节第 235 页
- [“UltraLite 初始化数据库实用程序 \(ulinit\)”](#) 一节第 249 页
- [“UltraLite 装载 XML 到数据库实用程序 \(ulload\)”](#) 一节第 252 页
- [“在 Palm OS 上保存、检索和清除加密密钥”](#) 一节 《UltraLite - C 及 C++ 编程》
- UltraLite for embedded SQL: [“ULCreateDatabase 函数”](#) 一节 《UltraLite - C 及 C++ 编程》
- UltraLite C/C++: [“ULChangeEncryptionKey 函数”](#) 一节 《UltraLite - C 及 C++ 编程》
- UltraLite for C++: [“CreateDatabase 函数”](#) 一节 《UltraLite - C 及 C++ 编程》
- UltraLite.NET: [“CreateDatabase 方法”](#) 一节 《UltraLite - .NET 编程》
- UltraLite.NET: [“ChangeEncryptionKey 方法”](#) 一节 《UltraLite - .NET 编程》
- UltraLite for M-Business Anywhere: [“createDatabase 方法”](#) 一节 《UltraLite - M-Business Anywhere 编程》
- UltraLite for M-Business Anywhere: [“changeEncryptionKey 方法”](#) 一节 《UltraLite - M-Business Anywhere 编程》
- [“访问创建参数值”](#) 一节第 30 页

## UltraLite max\_hash\_size 创建参数

以字节为单位设置最大的缺省索引散列大小。

### 语法

```
{ ulcreate | ulinit | ulload } -o max_hash_size=value;...
```

### 允许值

0 到 32 个字节

### 缺省值

4 个字节

### 注释

散列是存储在索引页中的索引条目的可选部分。散列会将索引列的实际行值转换为等效的数字值（键），同时仍然保留该索引的顺序。键大小以及 UltraLite 会散列实际值的多少，由所设置的散列大小确定。

行 ID 使得 UltraLite 可以定位实际数据在表中所处的行。行 ID 始终是索引条目的一部分。如果将散列大小设置为 0（禁用索引散列），则索引条目只包含此行 ID。对于所有其它散列大小，散列键（可以包含该行的所有或部分转换的数据）可以与行 ID 一起存储在索引页中。您可以在这些索引列上提高查询性能，因为在比较实际行值之前，UltraLite 可能并不总是需要查找、装载和解开数据。

确定一个恰当的缺省数据库散列大小，需要在查询效率和数据库大小之间进行折衷：最大散列值越高，数据库大小就会变得越大。

列的数据类型需要多少个字节，UltraLite 就使用多少个字节，但不超过此参数指定的最大值。仅当您在创建索引时没有设置大小的情况下才会使用缺省散列大小。如果将缺省散列大小设置为 0，则 UltraLite 不会散列行值。

不能更改现有索引的散列大小。当创建一个新索引时，可以使用 Sybase Central 中的 UltraLite 的 [\[创建索引向导\]](#) 或者 CREATE INDEX 或 CREATE TABLE 语句中的 WITH MAX SIZE 子句覆盖缺省值。

如果将列声明为 DOUBLE、FLOAT 或 REAL，则不使用散列。将始终忽略散列大小。

从 Sybase Central 中，可以在任何一个创建数据库向导中设置最大散列大小。在 [\[新数据库存储设置\]](#) 页面上，选择 [\[索引的最大散列大小\]](#) 选项。

在客户端应用程序中，对于数据库管理器类，将此参数设置为创建数据库方法的创建参数之一。

### 另请参见

- “UltraLite 性能与优化” 第 99 页
- “使用 UltraLite 索引” 一节第 71 页
- “选择最佳的散列大小” 一节第 105 页
- “UltraLite CREATE INDEX 语句” 一节第 433 页
- “UltraLite CREATE TABLE 语句” 一节第 440 页
- “UltraLite 创建数据库实用程序 (ulcreate)” 一节第 235 页
- “UltraLite 初始化数据库实用程序 (ulinit)” 一节第 249 页
- “UltraLite 装载 XML 到数据库实用程序 (ulload)” 一节第 252 页
- UltraLite for embedded SQL: “ULCreateDatabase 函数” 一节 《UltraLite - C 及 C++ 编程》
- UltraLite for C++: “CreateDatabase 函数” 一节 《UltraLite - C 及 C++ 编程》
- UltraLite.NET: “CreateDatabase 方法” 一节 《UltraLite - .NET 编程》
- UltraLite for M-Business Anywhere: “createDatabase 方法” 一节 《UltraLite - M-Business Anywhere 编程》
- “访问创建参数值” 一节第 30 页

## UltraLite nearest\_century 创建参数

控制字符串到日期的转换中对两位的年份的解释。

### 语法

```
{ ulcreate | ulinit | ulload } -o nearest_century=value;...
```

### 允许值

0 到 100（含）之间的整数

### 缺省值

50

### 注释

当需要日期值时，即使年份在字符串中仅用两位来表示，UltraLite 也会自动将字符串转换为日期。对于两位日期，需要设置适当的翻转值。小于该值的两位数年份转换为 20yy，而大于或等于该值的年份转换为 19yy。

选择适当的翻转值通常由以下因素决定：

- **使用两位日期** 否则，最接近世纪值转换不适用。小于所设置 nearest\_century 值的两位年份将转换为 20yy，而大于或等于该值的年份将转换为 19yy。  
建议您存储四位日期，以避免转换不正确的问题。请参见“[使用明确的日期和时间](#)”一节《[SQL Anywhere 服务器 - SQL 参考](#)》。
- **统一数据库兼容性** 例如，在以前，SQL Anywhere 的行为是将年份加上 1900。Adaptive Server Enterprise 的行为是使用最接近的世纪值，所以对于 yy 值小于 50 的任何年份，都会将年份设置为 20yy。
- **日期表示什么：过去的事件还是将来的事件** 出生年份由于发生在过去，通常是需较低翻转值的年份。所以对于 yy 小于 20 的任何年份，都应将年份设置为 20yy。但是，如果日期用作到期日期，则使用较高翻转值将是合理的选择，因为该日期发生在将来。

您不能更改现有数据库的最接近的世纪值。而是必须创建一个新数据库。

从 Sybase Central 中，可以在任何一个创建数据库向导中配置最接近的世纪值设置。在 [\[新数据库创建参数\]](#) 页面上，选择 [\[最接近的世纪值\]](#) 选项。

在客户端应用程序中，对于数据库管理器类，将此参数设置为创建数据库方法的创建参数之一。

### 另请参见

- “不明确的字符串到日期的转换”一节 《SQL Anywhere 服务器 - SQL 参考》
- “UltraLite 创建数据库实用程序 (ulcreate)”一节第 235 页
- “UltraLite 初始化数据库实用程序 (ulinit)”一节第 249 页
- “UltraLite 装载 XML 到数据库实用程序 (ulload)”一节第 252 页
- UltraLite for embedded SQL: “ULCreateDatabase 函数”一节 《UltraLite - C 及 C++ 编程》
- UltraLite for C++: “CreateDatabase 函数”一节 《UltraLite - C 及 C++ 编程》
- UltraLite.NET: “CreateDatabase 方法”一节 《UltraLite - .NET 编程》
- UltraLite for M-Business Anywhere: “createDatabase 方法”一节 《UltraLite - M-Business Anywhere 编程》
- “访问创建参数值”一节第 30 页

## UltraLite obfuscate 创建参数

控制着数据库中数据的模糊处理。模糊处理是一种简单的加密方式。

### 语法

```
{ ulcreate | ulinit | ulload } -o obfuscate=value;...
```

### 允许值

布尔值。

### 缺省值

0（不会对数据库进行模糊处理）

### 注释

简单加密等效于模糊处理，对于通过使用磁盘实用程序查看文件来破译数据库中数据的人而言，它会增大破译的难度。简单加密不要求使用密钥就可以对数据库加密。

如果您想将数据库设置为没有正确的加密密钥就不能访问，则必须使用高度加密。请参见“[UltraLite fips 创建参数](#)”一节第 176 页。

从 Sybase Central 中，可以在任何一个创建数据库向导中设置 obfuscation。在 [\[新数据库存储设置选项\]](#) 页面上，选择 [\[使用简单加密 \(Obfuscation\)\]](#) 选项。

在客户端应用程序中，对于数据库管理器类，将此参数设置为创建数据库方法的创建参数之一。

### 另请参见

- “简单加密”一节《[SQL Anywhere 服务器 - 数据库管理](#)》
- “保护 UltraLite 数据库”一节第 35 页
- “[UltraLite fips 创建参数](#)”一节第 176 页
- “[UltraLite DBKEY 连接参数](#)”一节第 214 页
- “[UltraLite 创建数据库实用程序 \(ulcreate\)](#)”一节第 235 页
- “[UltraLite 初始化数据库实用程序 \(ulinit\)](#)”一节第 249 页
- “[UltraLite 装载 XML 到数据库实用程序 \(ulload\)](#)”一节第 252 页
- UltraLite for embedded SQL: “[ULCreateDatabase 函数](#)”一节《[UltraLite - C 及 C++ 编程](#)》
- UltraLite for C++: “[CreateDatabase 函数](#)”一节《[UltraLite - C 及 C++ 编程](#)》
- UltraLite.NET: “[CreateDatabase 方法](#)”一节《[UltraLite - .NET 编程](#)》
- UltraLite for M-Business Anywhere: “[createDatabase 方法](#)”一节《[UltraLite - M-Business Anywhere 编程](#)》
- “[访问创建参数值](#)”一节第 30 页

## UltraLite page\_size 创建参数

定义数据库页面大小。

### 语法

```
{ ulcreate | ulinit | ulload } -o page_size=size[ k];...
```

### 允许值

1k、2k、4k、8k、16k

### 缺省值

4k

### 注释

UltraLite 数据库以页进行存储，所有 I/O 操作都是一次一页执行的。选择的页面大小会影响数据库的性能或大小。

如果使用所列值之外的任何其它值，则大小将变为下一个较大的页面大小。如果未指定单位，则假定以字节为单位。

如果您的平台拥有有限的动态内存，则考虑使用较小的页面大小以限制对同步内存需求的影响。

当选择了页面大小，则应该记住以下原则：

- **数据库大小** 较大的页面大小通常对于较大的数据库更有益。较大页能够保存更多信息，因此可以更有效地使用空间—特别是当您插入的行略大于页面大小的一半。页面越大，所需的页面交换越少。
- **行数** 由于一行（不包括 BLOB）必须放在一页上，所以页面大小决定了压缩行最大能有多大以及每页能存储多少行。在一些情况下，如果读取一页以获取一行的值，则可能会产生副作用：会将下几行的内容装入内存。请参见“[行压缩和表定义](#)”一节第 62 页。
- **查询类型** 通常，如果页面大小较小，可能有益于那些从任意位置检索较少行的操作。与之相对比，较大的页面往往对执行顺序表扫描的查询有利。
- **高速缓存大小** 大的页面大小可能需要较大的高速缓存大小。若高速缓存无法存放足够的页，在 UltraLite 开始与磁盘交换频繁使用的页时，性能就会受到影响。请参见“[UltraLite CACHE\\_SIZE 连接参数](#)”一节第 208 页。
- **索引条目** 页面大小也会影响索引。数据库页面越大，其中可以存放的索引条目就越多。请参见“[使用 UltraLite 索引](#)”一节第 71 页。
- **设备内存** 当数据库必须在内存有限的小型设备上运行时，小页尤为有用。例如，1 MB 的内存可以保存 1000 个大小均为 1 KB 的页，但只能保存 250 个大小均为 4 KB 的页。

您不能更改现有数据库的页大小。而是必须创建一个新数据库。

从 Sybase Central 中，可以在任何一个创建数据库向导中设置页面大小。在 [新数据库存储设置] 页面上，选择恰当的字节值。

在客户端应用程序中，对于数据库管理器类，将此参数设置为创建数据库方法的创建参数之一。



## 另请参见

- “UltraLite 优化策略” 一节第 112 页
- “行压缩和表定义” 一节第 62 页
- “UltraLite 大小写创建参数” 一节第 167 页
- “UltraLite CACHE\_SIZE 连接参数” 一节第 208 页
- “UltraLite RESERVE\_SIZE 连接参数” 一节第 224 页
- “UltraLite 创建数据库实用程序 (ulcreate)” 一节第 235 页
- “UltraLite 初始化数据库实用程序 (ulinit)” 一节第 249 页
- “UltraLite 装载 XML 到数据库实用程序 (ulload)” 一节第 252 页
- UltraLite for embedded SQL: “ULCreateDatabase 函数” 一节 《UltraLite - C 及 C++ 编程》
- UltraLite for C++: “CreateDatabase 函数” 一节 《UltraLite - C 及 C++ 编程》
- UltraLite.NET: “CreateDatabase 方法” 一节 《UltraLite - .NET 编程》
- UltraLite for M-Business Anywhere: “createDatabase 方法” 一节 《UltraLite - M-Business Anywhere 编程》
- “访问创建参数值” 一节第 30 页

## 示例

要将数据库的页面大小设置为 8 KB, 指定 **page\_size=8k** 或 **page\_size=8192**:

```
ulcreate test.udb -o page_size=8k
```

## UltraLite 精度创建参数

指定小数点算术结果中的最多位数。

### 语法

```
{ ulcreate | ulinit | ulload } -o precision=value;...
```

### 允许值

1 到 127（含）之间的整数

### 缺省值

30

### 注释

小数点位置由数字的精度和小数位数确定：精度是小数点左右的总位数；小数位数是按最大精度截断运算结果时小数点后的最小位数。

选择适当的小数点位置通常由以下因素决定：

- **所执行的运算过程类型** 乘、除、加、减和集合函数都可以有超过最大精度的结果。  
例如，当 DECIMAL(8,2) 与 DECIMAL(9,2) 相乘时，结果可能要求 DECIMAL(17,4)。如果精度为 15，则在结果中将只保留 15 位数字。如果小数位数为 4，则结果为 DECIMAL(15.4)。如果小数位数为 2，则结果为 DECIMAL(15.2)。两种情况下都有可能出现溢出错误。
- **小数位数值与精度值之间的关系** 小数位数设置数字的小数部分的位数，不能为负数，也不能大于精度。

您不能更改现有数据库的精度。而是必须创建一个新数据库。

如果使用 Oracle 数据库作为统一数据库，则所有 UltraLite 远程数据库和所有 Oracle 统一数据库必须拥有同一个精度值。

从 Sybase Central 中，可以在任何一个创建数据库向导中设置精度。在 [新数据库创建参数] 页面上，选择 [精度] 选项。

在客户端应用程序中，对于数据库管理器类，将此参数设置为创建数据库方法的创建参数之一。

### 另请参见

- “UltraLite 小数位数创建参数” 一节第 185 页
- “UltraLite 创建数据库实用程序 (ulcreate)” 一节第 235 页
- “UltraLite 初始化数据库实用程序 (ulinit)” 一节第 249 页
- “UltraLite 装载 XML 到数据库实用程序 (ulload)” 一节第 252 页
- UltraLite for embedded SQL: “ULCreateDatabase 函数” 一节 《UltraLite - C 及 C++ 编程》
- UltraLite for C++: “CreateDatabase 函数” 一节 《UltraLite - C 及 C++ 编程》
- UltraLite.NET: “CreateDatabase 方法” 一节 《UltraLite - .NET 编程》
- UltraLite for M-Business Anywhere: “createDatabase 方法” 一节 《UltraLite - M-Business Anywhere 编程》
- “访问创建参数值” 一节第 30 页

## UltraLite 小数位数创建参数

指定算术结果被截断为最大精度值时小数点后的最小位数。

### 语法

```
{ ulcreate | ulinit | ulload } -o scale=value;...
```

### 允许值

0 到 127（含）之间的整数

### 缺省值

6

### 注释

小数点位置由数字的精度和小数位数确定：精度是小数点左右的总位数；小数位数是按最大精度截断运算结果时小数点后的最小位数。

选择适当的小数点位置通常由以下因素决定：

- **所执行的运算过程类型** 乘、除、加、减和集合函数都可以有超过最大精度的结果。  
例如，当 DECIMAL(8,2) 与 DECIMAL(9,2) 相乘时，结果可能要求 DECIMAL(17,4)。如果精度为 15，则在结果中将只保留 15 位数字。如果小数位数为 4，则结果为 DECIMAL(15.4)。如果小数位数为 2，则结果为 DECIMAL(15.2)。两种情况下都有可能出现溢出错误。
- **小数位数值与精度值之间的关系** 小数位数设置数字的小数部分的位数，不能为负数，也不能大于精度。

您不能更改现有数据库的小数位数。而是必须创建一个新数据库。

从 Sybase Central 中，可以在任何一个创建数据库向导中设置小数位数。在 [\[新数据库创建参数\]](#) 页面上，选择 [\[范围\]](#) 选项。

在客户端应用程序中，对于数据库管理器类，将此参数设置为创建数据库方法的创建参数之一。

### 另请参见

- [“UltraLite 精度创建参数”](#) 一节第 184 页
- [“UltraLite 创建数据库实用程序 \(ulcreate\)”](#) 一节第 235 页
- [“UltraLite 初始化数据库实用程序 \(ulinit\)”](#) 一节第 249 页
- [“UltraLite 装载 XML 到数据库实用程序 \(ulload\)”](#) 一节第 252 页
- UltraLite for embedded SQL: [“ULCreateDatabase 函数”](#) 一节 《UltraLite - C 及 C++ 编程》
- UltraLite for C++: [“CreateDatabase 函数”](#) 一节 《UltraLite - C 及 C++ 编程》
- UltraLite.NET: [“CreateDatabase 方法”](#) 一节 《UltraLite - .NET 编程》
- UltraLite for M-Business Anywhere: [“createDatabase 方法”](#) 一节 《UltraLite - M-Business Anywhere 编程》
- [“访问创建参数值”](#) 一节第 30 页

### 示例

当 DECIMAL(8,2) 与 DECIMAL(9,2) 相乘时，结果可能要求 DECIMAL(17,4)。如果精度为 15，则在结果中将只保留 15 位数字。如果小数位数为 4，则结果为 DECIMAL(15,4)。如果小数位数为 2，则结果为 DECIMAL(15,2)。在这两个示例中都有溢出的可能。

## UltraLite time\_format 创建参数

为从数据库中检索得到的时间设置格式。

### 语法

```
{ ulcreate | ulinit | ulload } -o time_format=value;...
```

### 允许值

由下面所列的符号组成的字符串

### 缺省值

*HH:NN:SS.sss*

### 注释

UltraLite 会通过您使用 time\_format 创建参数设置的时间部分来写出时间。时间部分可包括小时、分钟、秒和毫秒。

也可以用字符串表示时间值。在时间值可以被检索之前，必须将其指派给一个字符串变量。

ISO (HH:MM:SS) 是缺省时间格式。例如，以此国际标准格式表示的 "午夜" 会写为：00:00:00。如果您不想使用缺省 ISO 时间格式，必须为这些时间部分指定另一不同的格式和顺序。

格式为使用以下符号的字符串：

符号	说明
<i>HH</i>	两位数表示的小时数（24 小时制）。
<i>NN</i>	两位数表示的分钟数。
<i>MM</i>	两位数表示的分钟数，如果跟在冒号后面（例如 <i>hh:mm</i> ）。
<i>SS[s...]</i>	两位的秒数加可选的小数部分。

您不能更改现有数据库的时间格式。而是必须创建一个新数据库。

每个符号都会替换为用该格式设置的相应时间值。任何表示字符而不是数字输出的格式符号都可以用大写输入，这样会使替换后的字符显示为大写。对于数字，在格式字符串中使用混合的大小写会取消前导零的显示。

可以根据符号的大小写情况来控制是否填充零：

- 以相同的大小写形式（例如 HH 或 hh）输入符号，可允许以零填充。例如，HH:NN:SS 可生成 01:01:01 形式的输出。
- 以大小写混合的形式（例如 Hh 或 hH）输入字符会取消以零填充。例如，Hh:Nn:Ss 可生成 1:1:1 形式的输出。

从 Sybase Central 中，可以在任何一个创建数据库向导中设置时间格式。在 [\[新数据库创建参数\]](#) 页面上，选择 [\[时间格式\]](#) 选项。

在客户端应用程序中，对于数据库管理器类，将此参数设置为创建数据库方法的创建参数之一。

### 另请参见

- “UltraLite timestamp\_format 创建参数” 一节第 189 页
- “UltraLite 创建数据库实用程序 (ulcreate)” 一节第 235 页
- “UltraLite 初始化数据库实用程序 (ulinit)” 一节第 249 页
- “UltraLite 装载 XML 到数据库实用程序 (ulload)” 一节第 252 页
- UltraLite for embedded SQL: “ULCreateDatabase 函数” 一节 《UltraLite - C 及 C++ 编程》
- UltraLite for C++: “CreateDatabase 函数” 一节 《UltraLite - C 及 C++ 编程》
- UltraLite.NET: “CreateDatabase 方法” 一节 《UltraLite - .NET 编程》
- UltraLite for M-Business Anywhere: “createDatabase 方法” 一节 《UltraLite - M-Business Anywhere 编程》
- “访问创建参数值” 一节第 30 页

### 示例

如果在 3:30 P.M. 执行了一个事务，且使用的缺省 time\_format 语法为 *HH:NN:SS.sss*，则结果应为：  
15:30:55.0

## UltraLite timestamp\_format 创建参数

为从数据库检索得来的时间戳设置格式。

### 语法

```
{ ulcreate | ulinit | ulload } -o timestamp_format=value;...
```

### 允许值

String

### 缺省值

YYYY-MM-DD HH:NN:SS.SSS

### 注释

UltraLite 会通过您使用 `date_format` 和 `time_format` 创建参数设置的日期和时间部分创建时间戳。日期和时间合在一起总共有七个部分（年、月、日、小时、分钟、秒和毫秒）。

也可以用字符串表示时间戳值。必须先将时间戳值赋给字符串变量，才能检索该值。

时间戳列通常用于确保在与统一数据库同步时保持数据完整性。通过跟踪每个用户最后一次同步的时间，可借助时间戳来确定何时在多个远程数据库之间进行了并发数据更新。

#### 提示

确保统一数据库和 UltraLite 远程数据库将时间戳和时间戳增量保持到相同的分辨率。通过设置这些创建参数，使其与统一数据库的属性相符，可帮助避免虚假的不相等。

格式为使用以下符号的字符串：

符号	说明
<i>YY</i>	两位数表示的年份。
<i>YYYY</i>	四位数表示的年份。
<i>MM</i>	两位数表示的月份，如果跟在冒号后面（例如， <i>hh:mm</i> ），则为两位数表示的分钟数。
<i>MMM[m...]</i>	月份的字符简写形式—有多少个“m”，就有多少个字符。大写字母 M 会使输出也是大写字母。
<i>D</i>	一位数表示的周中的某一天（0 代表星期日，6 代表星期六）。
<i>DD</i>	两位数表示的月中的某一天。不需要前导零。
<i>DDD[d...]</i>	周中某天的字符简写形式。大写字母 D 会使输出也是大写字母。
<i>HH</i>	两位数表示的小时数。不需要前导零。

符号	说明
<i>NN</i>	两位数表示的分钟数。不需要前导零。
<i>SS</i> [.ss..]	秒及秒的小数部分。
<i>AA</i>	使用 12 小时制。用 AM 表示上午的时间。
<i>PP</i>	使用 12 小时制。用 PM 表示下午的时间。
<i>JJJ</i>	年中的某一天，从 1 到 366。

您不能更改现有数据库的时间戳格式。而是必须创建一个新数据库。

允许的值是用上表中列出的符号构造的。每个符号都会替换为按此设置格式的相应日期值。

对于字符简写形式，计算指定的字母数。如有必要，还可以将 A.M. 或 P.M. 指示符（可以本地化）截断为与指定的字符数相应的字节数。

对于表示字符数据的符号（例如 *mmm*），可以控制输出的大小写，如下所示：

- 以全部大写的形式输入符号，可使格式以全部大写的形式显示。例如，MMM 会生成 JAN 形式的输出。
- 以全部小写的形式输入符号，可使格式以全部小写的形式显示。例如，mmm 会生成 jan 形式的输出。
- 以大小写混合的形式输入符号，让 UltraLite 针对所用语言来选择适当的大小写。例如，以英语输入 Mmm 会生成 May 形式的输出，而在法语中会生成 mai 形式的输出。

对于表示数字数据的符号，您可以根据符号的大小写情况来控制是否填充零：

- 以同样的大小写形式（例如 MM 或 mm）输入符号，可允许以零填充。例如，yyyy/mm/dd 可生成 2002/01/01 形式的输出。
- 以大小写混合的形式（例如 Mm）输入字符会取消以零填充。例如，yyyy/Mm/Dd 可生成 2002/1/1 形式的输出。

从 Sybase Central 中，可以在任何一个创建数据库向导中设置时间戳格式。在 [\[新数据库创建参数\]](#) 页面上，选择 [\[时间戳格式\]](#) 选项。

在客户端应用程序中，对于数据库管理器类，将此参数设置为创建数据库方法的创建参数之一。



## 另请参见

- “UltraLite timestamp\_increment 创建参数” 一节第 192 页
- “UltraLite 创建数据库实用程序 (ulcreate)” 一节第 235 页
- “UltraLite 初始化数据库实用程序 (ulinit)” 一节第 249 页
- “UltraLite 装载 XML 到数据库实用程序 (ulload)” 一节第 252 页
- UltraLite for embedded SQL: “ULCreateDatabase 函数” 一节 《UltraLite - C 及 C++ 编程》
- UltraLite for C++: “CreateDatabase 函数” 一节 《UltraLite - C 及 C++ 编程》
- UltraLite.NET: “CreateDatabase 方法” 一节 《UltraLite - .NET 编程》
- UltraLite for M-Business Anywhere: “createDatabase 方法” 一节 《UltraLite - M-Business Anywhere 编程》
- “访问创建参数值” 一节第 30 页
- “基于时间戳的下载” 一节 《MobiLink - 服务器管理》
- “UltraLite 中的并发” 一节第 11 页

## 示例

如果在 2006 年 5 月 12 日星期五的 3:30 PM 执行了一个事务，且您所使用的缺省 timestamp\_format 语法为 *YYYY-MM-DD HH:NN:SS.SSS*，则结果应为：

2006-05-12 15:30:55.0

## UltraLite timestamp\_increment 创建参数

限制时间戳值的精度。向数据库中插入时间戳时，UltraLite 会截断时间戳，使之与此增量匹配。

### 语法

```
{ ulcreate | ulinit | ulload } -o timestamp_increment=value;...
```

### 允许值

1 到 60000000 微秒

### 缺省值

1 微秒

### 注释

1000000 微秒等于 1 秒。

您不能更改现有数据库的时间戳增量。而是必须创建一个新数据库。

当使用某个 DEFAULT TIMESTAMP 列作为主键或行标识符时，此增量非常有用。

从 Sybase Central 中，可以在任何一个创建数据库向导中设置时间戳增量。在 [\[新数据库创建参数\]](#) 页面上，选择 [\[时间戳增量\]](#) 选项。

在客户端应用程序中，对于数据库管理器类，将此参数设置为创建数据库方法的创建参数之一。

### 另请参见

- [“UltraLite timestamp\\_format 创建参数”](#) 一节第 189 页
- [“UltraLite 创建数据库实用程序 \(ulcreate\)”](#) 一节第 235 页
- [“UltraLite 初始化数据库实用程序 \(ulinit\)”](#) 一节第 249 页
- [“UltraLite 装载 XML 到数据库实用程序 \(ulload\)”](#) 一节第 252 页
- UltraLite for embedded SQL: [“ULCreateDatabase 函数”](#) 一节 《UltraLite - C 及 C++ 编程》
- UltraLite for C++: [“CreateDatabase 函数”](#) 一节 《UltraLite - C 及 C++ 编程》
- UltraLite.NET: [“CreateDatabase 方法”](#) 一节 《UltraLite - .NET 编程》
- UltraLite for M-Business Anywhere: [“createDatabase 方法”](#) 一节 《UltraLite - M-Business Anywhere 编程》
- [“访问创建参数值”](#) 一节第 30 页
- [“基于时间戳的下载”](#) 一节 《MobiLink - 服务器管理》
- [“UltraLite 中的并发”](#) 一节第 11 页

### 示例

要存储诸如 '2000/12/05 10:50:53:700' 之类的值，将此创建参数设置为 100000。该值在秒部分中第一个小数位后截断时间戳。

## UltraLite utf8\_encoding 创建参数

使用 UTF-8 格式（Unicode 的 8 位多字节编码）对数据进行编码。

### 语法

```
{ ulcreate | ulinit | ulload } -o utf8_encoding=value;...
```

### 值

布尔值。

### 缺省值

0（数据库不是 UTF-8 编码形式）

### 注释

UTF-8 字符以 1 到 4 个字节表示。对于其它多字节归类，则使用一个或两个字节。对于提供的所有多字节归类，由两个或多个字节组成的字符被视为字母型。这意味着使用这些字符作标识符时无需双引号。

通过以 UTF-8 形式对数据库编码，UltraLite 使用 UTF8BIN 归类对字符排序。UTF8BIN 字符集并非特定于任何特定的本地语言；也没有具体的代码页与此字符集关联。因此，您可以将数据从多种本地语言同步为同一个统一数据库。如果试图将 UTF-8 编码字符同步为一个不支持 Unicode 的统一表，则会报告用户错误。

从 Sybase Central 中，可以在任何一个创建数据库向导中选择 UTF-8 编码。在 [\[新数据库归类和字符集\]](#) 页面，选择 [\[是，使用 UTF8 作为数据库字符集\]](#) 选项。

在客户端应用程序中，对于数据库管理器类，将此参数设置为创建数据库方法的创建参数之一。

### 另请参见

- [“字符集编码的 UltraLite 平台要求”一节第 33 页](#)
- [“UltraLite 字符集”一节第 32 页](#)
- [“UltraLite 创建数据库实用程序 \(ulcreate\)”一节第 235 页](#)
- [“UltraLite 初始化数据库实用程序 \(ulinit\)”一节第 249 页](#)
- [“UltraLite 装载 XML 到数据库实用程序 \(ulload\)”一节第 252 页](#)
- UltraLite for embedded SQL: [“ULCreateDatabase 函数”一节《UltraLite - C 及 C++ 编程》](#)
- UltraLite for C++: [“CreateDatabase 函数”一节《UltraLite - C 及 C++ 编程》](#)
- UltraLite.NET: [“CreateDatabase 方法”一节《UltraLite - .NET 编程》](#)
- UltraLite for M-Business Anywhere: [“createDatabase 方法”一节《UltraLite - M-Business Anywhere 编程》](#)
- [“访问创建参数值”一节第 30 页](#)

---

# UltraLite 数据库属性

## 目录

访问 UltraLite 数据库属性 .....	199
--------------------------	-----

属性	说明
<b>case</b>	返回区分大小写功能的状态。如果数据库区分大小写，则返回 <b>On</b> 。否则，返回 <b>Off</b> 。此属性的值在创建数据库时进行设置，并只能通过创建一个新数据库来更改。请参见 <a href="#">“UltraLite 大小写创建参数”</a> 一节第 167 页。
<b>char_set</b>	返回数据库的 CHAR 字符集。数据库所使用的字符集由数据库的归类序列以及数据是否为 UTF-8 编码格式来决定。  另请参见： <ul style="list-style-type: none"><li>● <a href="#">“UltraLite utf8_encoding 创建参数”</a> 一节第 193 页</li><li>● <a href="#">“UltraLite 归类创建参数”</a> 一节第 170 页</li></ul> 此属性的值在创建数据库时进行设置，并只能通过创建一个新数据库来更改。
<b>checksum_level</b>	返回数据库中校验和校验的级别：0（未添加校验和）、1（将校验和只添加到重要页面）或 2（将校验和添加到所有页面）之一。此属性的值在创建数据库时进行设置，并只能通过创建一个新数据库来更改。请参见 <a href="#">“UltraLite checksum_level 创建参数”</a> 一节第 168 页。
<b>collation</b>	返回数据库的归类序列名称。此属性的值在创建数据库时进行设置，并只能通过创建一个新数据库来更改。请参见 <a href="#">“UltraLite 归类创建参数”</a> 一节第 170 页。
<b>commit_flush_count</b>	返回设置提交计数阈值的 <code>commit_flush_count</code> 选项的值。请参见 <a href="#">“UltraLite commit_flush_count 选项 [临时性]”</a> 一节第 202 页。

属性	说明
<b>commit_flush_timeout</b>	返回设置时间间隔阈值的 <code>commit_flush_timeout</code> 选项的值。请参见“ <a href="#">UltraLite commit_flush_timeout 选项 [临时性]</a> ”一节第 203 页。
<b>conn_count</b>	返回数据库的连接数。值是动态的：它会随着当前存在的连接数而发生改变。UltraLite 最多支持十四个并发数据库连接。
<b>date_format</b>	返回数据库进行字符串转换所用的日期格式。此属性的值在创建数据库时进行设置，并只能通过创建一个新数据库来更改。请参见“ <a href="#">UltraLite date_format 创建参数</a> ”一节第 171 页。
<b>date_order</b>	返回数据库进行字符串转换所用的日期顺序。此属性的值在创建数据库时进行设置，并只能通过创建一个新数据库来更改。请参见“ <a href="#">UltraLite date_order 创建参数</a> ”一节第 174 页。
<b>encryption</b>	<p>返回数据库加密的类型：None、Simple、AES 或 AES_FIPS 之一。</p> <p>数据库所使用的加密由以下因素来决定：是否配置了高度加密（AES 或 AES_FIPS）以及 DBKEY 创建参数或模糊处理（简单加密）。</p> <p>仅当此属性的值最初为 None 时（即，既没有使用 <code>fips</code>，也没有使用模糊处理），才能更改此属性，然后通过调用 API 的正确函数或方法，在 <code>Connection</code> 对象上指定新的加密密钥来更改加密密钥在这种情况下，该值会更改为 AES，因为在创建数据库之后，将无法设置 <code>fips</code> 创建参数。请参见：</p> <ul style="list-style-type: none"> <li>● UltraLite C/C++: “<a href="#">ULChangeEncryptionKey 函数</a>”一节《<a href="#">UltraLite - C 及 C++ 编程</a>》</li> <li>● UltraLite.NET: “<a href="#">ChangeEncryptionKey 方法</a>”一节《<a href="#">UltraLite - .NET 编程</a>》</li> <li>● UltraLite for M-Business Anywhere: “<a href="#">changeEncryptionKey 方法</a>”一节《<a href="#">UltraLite - M-Business Anywhere 编程</a>》</li> <li>● “<a href="#">保护 UltraLite 数据库</a>”一节第 35 页</li> <li>● “<a href="#">UltraLite fips 创建参数</a>”一节第 176 页</li> <li>● “<a href="#">UltraLite obfuscate 创建参数</a>”一节第 181 页</li> <li>● “<a href="#">UltraLite DBKEY 连接参数</a>”一节第 214 页</li> </ul>

属性	说明
<b>file</b>	<p>返回当前连接的数据库根文件的名称，包括路径。它是 DBF 连接参数值所指定的值。请参见：</p> <ul style="list-style-type: none"> <li>● UltraLite C/C++: “GetDatabaseProperty 函数” 一节 《UltraLite - C 及 C++ 编程》</li> <li>● UltraLite.NET: “GetDatabaseProperty 方法” 一节 《UltraLite - .NET 编程》</li> <li>● UltraLite for M-Business Anywhere: “getDatabaseProperty 方法” 一节 《UltraLite - M-Business Anywhere 编程》</li> <li>● “UltraLite DBF 连接参数” 一节第 212 页</li> </ul>
<b>global_database_id</b>	<p>返回用于全局自动增量列的 global_database_id 选项的值。请参见 “UltraLite global_database_id 选项” 一节第 204 页。</p>
<b>max_hash_size</b>	<p>返回用于检索散列的缺省最大字节数。可基于每个索引设置此属性。请参见 “UltraLite max_hash_size 创建参数” 一节第 177 页。</p>
<b>ml_remote_id</b>	<p>返回唯一标识数据库以实现 MobiLink 同步的 ml_remote_id 选项的值。请参见 “UltraLite ml_remote_id 选项” 一节第 205 页。</p>
<b>name</b>	<p>返回当前连接的数据库名称（或别名）。返回的名称与 DBN 连接参数值匹配。如果尚未使用 DBN 连接参数，返回的名称是不带路径和扩展名的数据库文件。</p> <p>另请参见：</p> <ul style="list-style-type: none"> <li>● “UltraLite DBN 连接参数” 一节第 215 页</li> <li>● “UltraLite DBF 连接参数” 一节第 212 页</li> </ul>
<b>nearest_century</b>	<p>返回数据库进行字符串转换所使用的最近的世纪值。此属性的值在创建数据库时进行设置，并只能通过创建一个新数据库来更改。请参见 “UltraLite nearest_century 创建参数” 一节第 179 页。</p>
<b>page_size</b>	<p>返回数据库的页大小（以字节为单位）。此属性的值在创建数据库时进行设置，并只能通过创建一个新数据库来更改。请参见 “UltraLite page_size 创建参数” 一节第 182 页。</p>
<b>precision</b>	<p>返回数据库进行字符串转换所使用的浮点精度。此属性的值在创建数据库时进行设置，并只能通过创建一个新数据库来更改。请参见 “UltraLite 精度创建参数” 一节第 184 页。</p>

属性	说明
<b>scale</b>	返回数据库进行字符串转换的过程中，算术结果按最大 PRECISION 值截断时小数点后的最少位数。此属性的值在创建数据库时进行设置，并只能通过创建一个新数据库来更改。请参见“ <a href="#">UltraLite 小数位数创建参数</a> ”一节第 185 页。
<b>time_format</b>	返回数据库进行字符串转换所使用的时间格式。此属性的值在创建数据库时进行设置，并只能通过创建一个新数据库来更改。请参见“ <a href="#">UltraLite time_format 创建参数</a> ”一节第 187 页。
<b>timestamp_format</b>	返回数据库进行字符串转换所使用的时间戳格式。此属性的值在创建数据库时进行设置，并只能通过创建一个新数据库来更改。请参见“ <a href="#">UltraLite timestamp_format 创建参数</a> ”一节第 189 页。
<b>timestamp_increment</b>	返回两个唯一时间戳之间的最小差异，以微秒为单位。此属性的值在创建数据库时进行设置，并只能通过创建一个新数据库来更改。请参见“ <a href="#">UltraLite timestamp_increment 创建参数</a> ”一节第 192 页。



## 访问 UltraLite 数据库属性

UltraLite 为您提供一组用于检索数据库的属性。

可以更改任何不与数据库创建参数对应的数据库属性的设置。

### ◆ 浏览 UltraLite 数据库属性 (Sybase Central)

1. 连接到数据库。
2. 右键单击数据库并选择 [属性]。

在 [数据库属性] 窗口中，数据库属性在 [常规] 和 [扩展信息] 选项卡上列出。在 [扩展信息] 选项卡上，数据库属性按属性名称的字母顺序列出。若要按值对数据库属性进行排序，请单击 [值] 列。

3. 如果在您开始浏览数据库属性后，它们发生了更改，请单击 [刷新]。

### ◆ 获得数据库属性的值 (C/C++)

- 用 C/C++ 调用 `GetDatabaseProperty` 函数。

例如，要获得 `conn_count` 属性的值，调用：

```
GetDatabaseProperty( ul_database_property_id conn_count )
```

要获得 `char_set` 属性的值，调用：

```
GetDatabaseProperty( ul_database_property_id char_set )
```

### 另请参见

- UltraLite C/C++: “[GetDatabaseProperty 函数](#)”一节 《UltraLite - C 及 C++ 编程》
- UltraLite.NET: “[GetDatabaseProperty 方法](#)”一节 《UltraLite - .NET 编程》
- UltraLite for M-Business Anywhere: “[getDatabaseProperty 方法](#)”一节 《UltraLite - M-Business Anywhere 编程》

---

---

# UltraLite 数据库选项

## 目录

UltraLite commit_flush_count 选项 [临时性] .....	202
UltraLite commit_flush_timeout 选项 [临时性] .....	203
UltraLite global_database_id 选项 .....	204
UltraLite ml_remote_id 选项 .....	205
变更 UltraLite 持久性数据库选项设置 .....	206

---

用于配置数据库行为的选项。可以在任何时候对数据库选项进行设置或修改。在 UltraLite 中，选项可以是持久的，也可以是临时的。持久选项存储在数据库中的 `sysuldata` 系统表中。临时选项设置仅在数据库运行时被保留。

通过使用 SET OPTION 语句设置选项值。例如，以下语句将 `global_database_id` 选项设置为 100:

```
SET OPTION global_database_id=100;
```

可以通过查询相应数据库属性的值或通过使用适当的获取数据库属性的方法来获取数据库选项的当前设置。例如，要获取 `commit_flush_timeout` 数据库选项的当前设置，请执行如下 SQL 语句:

```
SELECT DB_PROPERTY ( 'commit_flush_timeout' );
```

## UltraLite commit\_flush\_count 选项 [临时性]

设置一个提交计数阈值，之后执行一次提交刷新。

### 允许值

整数

### 缺省值

10

### 注释

使用 0 禁用事务计数。在禁用事务计数的情况下，当触发刷新时，提交的数目没有限制。

每此您启动数据库时必须设置此选项（如果需要）。

commit\_flush\_count 和 commit\_flush\_timeout 都是临时数据库选项。每次启动数据库时都必须设置这些选项。只要数据库继续运行，它们就会一直保留。仅当将 COMMIT\_FLUSH=grouped 设置为连接字符串的一部分时，才需要这两个选项。

在设置此选项并将 COMMIT\_FLUSH 连接参数设置为在连接字符串中分组时，任一个阈值都会触发刷新。刷新后，UltraLite 将计数器 和 计时器清零。然后，同时监控计数器和计时器，直到再一次达到这些阈值。

设置提交刷新选项时需要考虑的一个重要事项是，刷新已提交事务的延迟会给数据恢复造成多大的风险。稍有机会事务就可能丢失即使它已经提交。在提交之后而在将事务刷新到存储器之前，如果发生严重的硬件故障，则会在恢复时回退该事务。较长的延迟可以提高 UltraLite 性能。您必须谨慎选择适当的计数阈值。

要从客户端应用程序设置 commit\_flush\_count 选项，则使用设置数据库选项功能为您所使用的编程接口设置此选项。

### 另请参见

- [“刷新单个或分组的事务”一节第 110 页](#)
- [“UltraLite commit\\_flush\\_timeout 选项 \[临时性\]”一节第 203 页](#)
- [“UltraLite COMMIT\\_FLUSH 连接参数”一节第 210 页](#)
- [“UltraLite SET OPTION 语句”一节第 459 页](#)
- UltraLite for C/C++: [“SetDatabaseOption 函数”一节 《UltraLite - C 及 C++ 编程》](#)
- UltraLite for embedded SQL: [“ULSetDatabaseOptionString 函数”一节 《UltraLite - C 及 C++ 编程》](#)

## UltraLite commit\_flush\_timeout 选项 [临时性]

设置一个时间间隔阈值，之后执行一次分组提交刷新。

### 允许值

整数（以毫秒为单位）

### 缺省值

10000 毫秒

### 注释

使用 0 禁用时间阈值。

每次启动数据库时必须设置此选项，如果需要。

commit\_flush\_count 和 commit\_flush\_timeout 都是临时数据库选项。每次启动数据库时都必须设置这些选项。只要数据库继续运行，它们就会一直保留。仅当将 COMMIT\_FLUSH=grouped 设置为连接字符串的一部分时，才需要这两个选项。

如果除了 commit\_flush\_timeout 选项外还设置了此选项，且将 COMMIT\_FLUSH 连接参数设置为 grouped，则这两个阈值中的任何一个都会触发刷新。刷新后，UltraLite 将计数器 和 计时器清零。然后，同时监控计数器和计时器，直到再一次达到这些阈值。

设置提交刷新选项时需要考虑的一个重要事项是，刷新已提交事务的延迟会给数据恢复造成多大的风险。稍有机会事务就可能丢失即使它已经提交。在提交之后而在将事务刷新到存储器之前，如果发生严重的硬件故障，则会在恢复时回退该事务。较长的延迟可以提高 UltraLite 性能。您必须谨慎选择适当的超时阈值。

要从客户端应用程序设置 commit\_flush\_timeout 选项，则使用设置数据库选项功能为您所使用的编程接口设置此选项。

### 另请参见

- [“UltraLite commit\\_flush\\_count 选项 \[临时性\]”一节第 202 页](#)
- [“UltraLite COMMIT\\_FLUSH 连接参数”一节第 210 页](#)
- [“UltraLite SET OPTION 语句”一节第 459 页](#)
- UltraLite for C/C++: [“SetDatabaseOption 函数”一节 《UltraLite - C 及 C++ 编程》](#)
- UltraLite for embedded SQL: [“ULSetDatabaseOptionString 函数”一节 《UltraLite - C 及 C++ 编程》](#)

## UltraLite global\_database\_id 选项

设置数据库标识号。

### 允许值

唯一的非负整数

### 缺省值

特定全局自动增量列的缺省值范围为  $pn + 1$  到  $p(n + 1)$ ，其中， $p$  为列的分区大小， $n$  为全局数据库标识号。

### 注释

为在与 MobiLink 服务器同步时保持主键唯一性，全局 ID 设置 GLOBAL AUTOINCREMENT 列的起始值。必须设置全局 ID 才能指派缺省值。如果向表中添加行后还没有为其设置值，UltraLite 会通过结合 global\_database\_id 值和分区大小来为该列生成一个值。请参见“使用全局自动增量”一节《MobiLink - 服务器管理》。

在部署应用程序时，必须为每个数据库指派不同的标识号，目的是为了与 MobiLink 服务器同步。可以随时更改现有数据库的全局 ID。

也可以使用 ulinfo 实用程序设置此选项：

```
ulinfo -g ID ...
```

要从客户端应用程序设置 global\_database\_id 选项，请为所使用的编程接口使用设置数据库 ID 功能。

### 另请参见

- “变更 UltraLite 持久性数据库选项设置”一节第 206 页
- “UltraLite 信息实用程序 (ulinfo)”一节第 245 页
- “UltraLite SET OPTION 语句”一节第 459 页
- “在 UltraLite 中使用 GLOBAL AUTOINCREMENT”一节第 119 页
- UltraLite for C/C++: “SetDatabaseID 函数”一节《UltraLite - C 及 C++ 编程》
- UltraLite for embedded SQL: “ULSetDatabaseID 函数”一节《UltraLite - C 及 C++ 编程》
- UltraLite.NET: “DatabaseID 属性”一节《UltraLite - .NET 编程》
- UltraLite for M-Business Anywhere: “setDatabaseID 方法”一节《UltraLite - M-Business Anywhere 编程》

### 示例

要使 UltraLite 数据库的列从 3001 自动增加到 4000，将全局 ID 设置为 3。

```
SET OPTION global_database_id="3";
```

## UltraLite ml\_remote\_id 选项

UltraLite 中用于 MobiLink 同步的唯一标识符。

### 允许值

用于唯一标识数据库以实现 MobiLink 同步的任何值。

### 缺省值

Null

### 注释

**远程 ID** 是用于进行 MobiLink 同步的 UltraLite 远程数据库的唯一标识符。MobiLink 远程 ID 初始情况下设置为 NULL。在第一次同步期间，MobiLink 服务器将远程 ID 设置为 GUID。不过，远程 ID 可以是任何有意义的字符串，只要该字符串在所有远程 MobiLink 客户端之间保持唯一。必须始终履行该唯一性要求。

远程 ID 存储 MobiLink 用户名的同步进度。通过加入唯一的远程 ID，就不再要求用户名唯一了。当有多个 MobiLink 用户与同一个 UltraLite 客户端数据库进行同步时，远程 ID 尤为有用。在这种情况下，同步脚本应该引用远程 ID，而不是只引用用户名。

也可以使用 ulinfo 实用程序设置此选项：

```
ulinfo -r ID ...
```

要从客户端应用程序设置 ml\_remote\_id 选项，则使用设置数据库选项功能为您所使用的编程接口设置此选项。

### 另请参见

- “变更 UltraLite 持久性数据库选项设置” 一节第 206 页
- “远程 ID” 一节 《MobiLink - 客户端管理》
- “授予 REMOTE 权限” 一节 《SQL Remote》
- “UltraLite 信息实用程序 (ulinfo)” 一节第 245 页
- “UltraLite SET OPTION 语句” 一节第 459 页
- UltraLite for C/C++: “SetDatabaseOption 函数” 一节 《UltraLite - C 及 C++ 编程》
- UltraLite for embedded SQL: “ULSetDatabaseOptionString 函数” 一节 《UltraLite - C 及 C++ 编程》
- UltraLite.NET: “SetDatabaseOption 方法” 一节 《UltraLite - .NET 编程》
- “UltraLite 客户端” 第 115 页
- “UltraLite 用户验证” 一节 《MobiLink - 客户端管理》
- “User Name 同步参数” 一节第 158 页
- 和 “Password 同步参数” 一节第 147 页

## 变更 UltraLite 持久性数据库选项设置

可以从 Sybase Central 查看和更改持久数据库选项的设置。临时 UltraLite 数据库选项无法从 Sybase Central 进行查看或设置。

### ◆ 浏览或修改持久 UltraLite 数据库选项 (Sybase Central)

1. 连接到数据库。
2. 右击数据库并选择 [选项]。
3. 如果您想要设置或重置某个选项，请在 [值] 字段中键入一个新值。
4. 单击 [立即设置] 或 [立即重置] 来提交更改。

### 另请参见

- [“UltraLite SET OPTION 语句”一节第 459 页](#)
- [“访问 UltraLite 数据库属性”一节第 199 页](#)
- [“DB\\_PROPERTY 函数 \[System\]”一节第 355 页](#)
- UltraLite C/C++: [“GetDatabaseProperty 函数”一节 《UltraLite - C 及 C++ 编程》](#)
- UltraLite.NET: [“GetDatabaseProperty 方法”一节 《UltraLite - .NET 编程》](#)
- UltraLite for M-Business Anywhere: [“getDatabaseProperty 方法”一节 《UltraLite - M-Business Anywhere 编程》](#)



---

# UltraLite 连接参数

## 目录

UltraLite CACHE_SIZE 连接参数 .....	208
UltraLite CE_FILE 连接参数 .....	209
UltraLite COMMIT_FLUSH 连接参数 .....	210
UltraLite CON 连接参数 .....	211
UltraLite DBF 连接参数 .....	212
UltraLite DBKEY 连接参数 .....	214
UltraLite DBN 连接参数 .....	215
UltraLite MIRROR_FILE 连接参数 .....	216
UltraLite NT_FILE 连接参数 .....	218
UltraLite ORDERED_TABLE_SCAN 连接参数 [不建议使用] .....	219
UltraLite PALM_ALLOW_BACKUP 连接参数 .....	220
UltraLite PALM_FILE 连接参数 .....	221
UltraLite PWD 连接参数 .....	223
UltraLite RESERVE_SIZE 连接参数 .....	224
UltraLite START 连接参数 .....	225
UltraLite UID 连接参数 .....	226

---

## UltraLite CACHE\_SIZE 连接参数

定义数据库高速缓存的大小。

### 语法

**CACHE\_SIZE**=number{ k | m | g }

### 缺省

缺省的高速缓存大小由系统上的可用内存量和数据库的大小确定。

### 注释

如果未指定高速缓存大小，或者如果将大小设置为 0，则使用缺省大小。如果测试显示需要提高性能，则应该增加高速缓存大小。

缺省情况下，大小的单位为字节。分别使用 k、m 或 g 将单位指定为千字节、兆字节或千兆字节。

如果指定值超过了最大高速缓存大小，将自动将该值替换为此平台高速缓存大小的上限。将高速缓存大小增大到超过数据库的大小不会带来任何性能上的提高，并且高速缓存过大可以影响能够使用的其它应用程序的数目。

参数值中的所有前导空格和尾随空格均被忽略。此连接参数的值不能包含前导单引号、前导双引号或分号。

### 另请参见

- “UltraLite 优化策略”一节第 112 页
- “使用连接字符串打开 UltraLite 连接”一节第 42 页
- “UltraLite page\_size 创建参数”一节第 182 页
- “UltraLite RESERVE\_SIZE 连接参数”一节第 224 页
- UltraLite for C/C++: “连接到数据库”一节 《UltraLite - C 及 C++ 编程》
- UltraLite for C/C++: “OpenConnection 函数”一节 《UltraLite - C 及 C++ 编程》
- UltraLite for embedded SQL: “连接到数据库”一节 《UltraLite - C 及 C++ 编程》
- UltraLite for M-Business Anywhere: “连接到 UltraLite 数据库”一节 《UltraLite - M-Business Anywhere 编程》
- UltraLite for M-Business Anywhere: “openConnection 方法”一节 《UltraLite - M-Business Anywhere 编程》
- UltraLite.NET: “连接到数据库”一节 《UltraLite - .NET 编程》
- UltraLite.NET: “Open 方法”一节 《UltraLite - .NET 编程》

### 示例

以下连接字符串代码段将高速缓存大小设置为 2 MB。

```
"CACHE_SIZE=2m"
```

## UltraLite CE\_FILE 连接参数

在创建数据库时命名新的数据库文件。打开与现有数据库的连接时，此参数标识该数据库。

### 语法

```
CE_FILE=path\ce-db
```

### 缺省值

- DBF 连接参数。
- 如果 DBF 连接参数或此连接参数的值均未设置，则缺省值为 `\UltraLiteDBtore.udb`。

### 注释

应为使用同一连接字符串的 UltraLite 客户端应用程序使用 CE\_FILE 连接参数，以连接到 Microsoft Windows Mobile 设备以及其它平台。

CE\_FILE 连接参数优先于 DBF 连接参数。如果要从 UltraLite 管理工具进行连接，或者连接对象只连接到 Windows Mobile 数据库，请使用 DBF 连接参数。

CE\_FILE 的值必须满足 Windows Mobile 的文件名要求。如果包含了数据库的绝对路径，则在设置此文件的路径之前，所有目录必须都已存在。UltraLite 不会自动创建目录。

参数值中的所有前导空格和尾随空格均被忽略。此连接参数的值不能包含前导单引号、前导双引号或分号。

### 另请参见

- [“使用连接字符串打开 UltraLite 连接”一节第 42 页](#)
- [“在 UltraLite 连接参数中指定文件路径”一节第 40 页](#)
- [“用于 UltraLite 管理工具的连接参数的优先级”一节第 43 页](#)
- [“UltraLite DBF 连接参数”一节第 212 页](#)
- UltraLite for C/C++: [“连接到数据库”一节《UltraLite - C 及 C++ 编程》](#)
- UltraLite for C/C++: [“OpenConnection 函数”一节《UltraLite - C 及 C++ 编程》](#)
- UltraLite for embedded SQL: [“连接到数据库”一节《UltraLite - C 及 C++ 编程》](#)
- UltraLite for M-Business Anywhere: [“连接到 UltraLite 数据库”一节《UltraLite - M-Business Anywhere 编程》](#)
- UltraLite for M-Business Anywhere: [“openConnection 方法”一节《UltraLite - M-Business Anywhere 编程》](#)
- UltraLite.NET: [“连接到数据库”一节《UltraLite - .NET 编程》](#)
- UltraLite.NET: [“Open 方法”一节《UltraLite - .NET 编程》](#)

### 示例

以下示例创建一个新连接，并为 Windows 桌面操作系统平台和 Windows Mobile 平台标识不同的数据库文件：

```
Set Connection = DatabaseMgr.OpenConnection("DBF=d:\Dbfile.udb;CE_FILE=\myapp\nMyDB.udb")
```

## UltraLite COMMIT\_FLUSH 连接参数

确定在提交调用后何时将提交的事务刷新到存储。如果 UltraLite 应用程序没有产生需要提交的调用，则不进行刷新。

### 语法

```
COMMIT_FLUSH={ immediate | grouped | on_checkpoint }
```

### Default

immediate

### 注释

此连接参数定义了硬件出现故障或崩溃后将恢复哪些事务。可以将多个逻辑自动提交操作分组为一个恢复点。

通过分组这些操作，可以提高 UltraLite 性能，但是却以牺牲数据可恢复性为代价。发生一种很少出现的情况时，事务会丢失，即使已经提交了事务—即在事务提交之后但尚未刷新到存储之前硬件出现了故障或崩溃。

支持以下参数：

- **immediate** 在提交操作完成之前，产生提交调用后就立即将提交的事务刷新到存储。
- **grouped** 产生提交调用且只在已达到所配置的阈值以后，将提交的事务刷新到存储。可以用 `commit_flush_count` 数据库选项配置事务计数的阈值，也可以用 `commit_flush_timeout` 数据库选项配置基于时间的阈值。

如果将 `commit_flush_count` 和 `commit_flush_timeout` 选项均设置为提交刷新的可用触发器，则首先达到的阈值将触发刷新。刷新后，UltraLite 将计数器和计时器清零。然后，同时监控计数器和计时器，直到再一次达到这些阈值之一。

- **on\_checkpoint** 执行检查点操作时将提交的事务刷新到存储。可以使用以下任一方法执行检查点：
  - CHECKPOINT 语句。没有 `checkpoint` 方法的 API 必须使用此 SQL 语句。
  - UltraLite 嵌入式 SQL 的 `ULCheckpoint` 函数。
  - C++ 组件中连接对象的 `Checkpoint` 方法。

### 另请参见

- “刷新单个或分组的事务”一节第 110 页
- “UltraLite `commit_flush_count` 选项 [临时性]”一节第 202 页
- “UltraLite `commit_flush_timeout` 选项 [临时性]”一节第 203 页
- “UltraLite CHECKPOINT 语句”一节第 431 页
- UltraLite 嵌入式 SQL：“`ULCheckpoint` 函数”一节 《UltraLite - C 及 C++ 编程》
- UltraLite C++：“`Checkpoint` 函数”一节 《UltraLite - C 及 C++ 编程》

## UltraLite CON 连接参数

对连接进行命名，以便在多连接应用程序中更容易地进行切换。

### 语法

**CON**=*name*

### 缺省值

无连接名。

### 注释

CON 连接参数对于应用程序来说是全局的。

除非要建立两个或更多的并发连接，并在它们之间进行切换，否则请勿使用此参数。

连接名与数据库名不同。

参数值中的所有前导空格和尾随空格均被忽略。此连接参数的值不能包含前导单引号、前导双引号或分号。

### 另请参见

- “使用连接字符串打开 UltraLite 连接” 一节第 42 页
- “UltraLite DBN 连接参数” 一节第 215 页
- UltraLite for C/C++: “连接到数据库” 一节 《UltraLite - C 及 C++ 编程》
- UltraLite for C/C++: “OpenConnection 函数” 一节 《UltraLite - C 及 C++ 编程》
- UltraLite for embedded SQL: “连接到数据库” 一节 《UltraLite - C 及 C++ 编程》
- UltraLite for M-Business Anywhere: “连接到 UltraLite 数据库” 一节 《UltraLite - M-Business Anywhere 编程》
- UltraLite for M-Business Anywhere: “openConnection 方法” 一节 《UltraLite - M-Business Anywhere 编程》
- UltraLite.NET: “连接到数据库” 一节 《UltraLite - .NET 编程》
- UltraLite.NET: “Open 方法” 一节 《UltraLite - .NET 编程》

### 示例

以下连接字符串代码段将第一个连接名设置为 MyFirstCon。

```
"CON=MyFirstCon"
```

## UltraLite DBF 连接参数

在创建数据库时命名新的数据库文件。打开连接时，此参数指示要装载并连接到的数据库文件。

### 语法

**DBF=ul-db**

### 缺省值

- **对于桌面操作系统平台** 如果未指定 NT\_FILE 或 DBF 值，则 UltraLite 将文件名设置为 *\UltraLiteDBstore.udb*。
- **在 Windows Mobile 上** 如果未指定 CE\_FILE 或 DBF 值，则 UltraLite 将文件名设置为 *\UltraLiteDBstore.udb*。
- **对于 Palm OS** 如果未指定 PALM\_FILE 或 DBF 值，则 UltraLite 将文件名设置为 *ulstore.udb*。

### 注释

如果要使用一个连接字符串连接到不同设备上的多个数据库，则可使用以下参数来命名特定于平台的替代参数：

- CE\_FILE
- PALM\_FILE
- NT\_FILE

如果已指定上述参数，则这些特定于平台的连接参数优先于 DBF。

DBF 的值必须满足所在平台的文件名要求。

**Palm OS** 您不能使用 Palm 安装工具将数据库部署到 VFS 卷。而是，您需要将数据库直接复制到带有读卡器的介质或某些其它工具上。

如果在使用 DBF 参数创建文件，则必须使用 *.pdb* 扩展名命名数据库。在桌面操作系统上管理 Palm OS 数据库时使用此数据库名称。管理工具和实用程序可以使用 DBF 连接参数连接到数据库，并且数据库具有 *.pdb* 扩展名。但是，一旦将此文件部署到设备，扩展名 *.pdb* 将被删除。随后您的应用程序必须使用 **Palm\_FILE** 参数连接到数据库，且数据库没有扩展名。

例如，如果您在管理工具桌面操作系统上使用 **DBF=CustDB.pdb** 命名文件 *CustDB.pdb*。则当数据库部署到设备时，应用程序使用 **PALM\_FILE=CustDB** 连接到同一数据库。

**Windows Mobile** 如果要部署到 Windows Mobile 设备，则 UltraLite 实用程序和向导可在附加的 Windows Mobile 设备上管理 UltraLite 数据库。要在 Windows Mobile 设备上标识文件，则必须指定所需的绝对路径，并使用 **wce:\** 前缀。

参数值中的所有前导空格和尾随空格均被忽略。该值不能包含前导单引号、前导双引号或分号。

## 另请参见

- “使用连接字符串打开 UltraLite 连接” 一节第 42 页
- “在 UltraLite 连接参数中指定文件路径” 一节第 40 页
- “用于 UltraLite 管理工具的连接参数的优先级” 一节第 43 页
- “UltraLite DBN 连接参数” 一节第 215 页
- “UltraLite CE\_FILE 连接参数” 一节第 209 页
- “UltraLite PALM\_FILE 连接参数” 一节第 221 页
- “UltraLite NT\_FILE 连接参数” 一节第 218 页
- UltraLite for C/C++: “连接到数据库” 一节 《UltraLite - C 及 C++ 编程》
- UltraLite for C/C++: “OpenConnection 函数” 一节 《UltraLite - C 及 C++ 编程》
- UltraLite for embedded SQL: “连接到数据库” 一节 《UltraLite - C 及 C++ 编程》
- UltraLite for M-Business Anywhere: “连接到 UltraLite 数据库” 一节 《UltraLite - M-Business Anywhere 编程》
- UltraLite for M-Business Anywhere: “openConnection 方法” 一节 《UltraLite - M-Business Anywhere 编程》
- UltraLite.NET: “连接到数据库” 一节 《UltraLite - .NET 编程》
- UltraLite.NET: “Open 方法” 一节 《UltraLite - .NET 编程》

## 示例

要连接到安装在桌面操作系统目录 `c:\mydb` 下的数据库 `MyULdb.udb`，请使用以下连接字符串：

```
"DBF=c:\mydb\MyULdb.udb"
```

要连接部署到附加的 Windows Mobile 设备的 *UltraLite* 文件夹中的同一数据库，请使用以下连接字符串：

```
"DBF=wce:\UltraLite\MyULdb.udb"
```

## UltraLite DBKEY 连接参数

创建新数据库时，为数据库提供加密密钥。打开与现有数据库的连接时，此连接参数为数据库提供加密密钥。

### 语法

**DBKEY**=*string*

### 缺省值

不提供密钥。

### 注释

如果没有为数据库指定正确的加密密钥，则连接失败。

如果使用加密密钥创建数据库，则会使用 AES 128 位或 AES FIPS 算法对数据库文件进行高度加密。使用高度加密可以提高安全性，从而防范手段高超的蓄意获取数据访问权限的攻击。但使用高度加密对性能也会产生显著的影响。

在 Palm OS 上，每当用户切换到其它应用程序，系统都会自动关闭原来的应用程序。不过，您可以对自己的 UltraLite 客户端进行编程，以避免每当用户再次切换回该应用程序时，需要重新输入密钥。

参数值中的所有前导空格和尾随空格均被忽略。该值不能包含前导单引号、前导双引号或分号。

### 另请参见

- UltraLite for C/C++: “在 Palm OS 上保存、检索和清除加密密钥”一节 《UltraLite - C 及 C++ 编程》
- “使用连接字符串打开 UltraLite 连接”一节第 42 页
- “保护 UltraLite 数据库”一节第 35 页
- “UltraLite obfuscate 创建参数”一节第 181 页
- UltraLite for C/C++: “连接到数据库”一节 《UltraLite - C 及 C++ 编程》
- UltraLite for C/C++: “OpenConnection 函数”一节 《UltraLite - C 及 C++ 编程》
- UltraLite for embedded SQL: “连接到数据库”一节 《UltraLite - C 及 C++ 编程》
- UltraLite for M-Business Anywhere: “连接到 UltraLite 数据库”一节 《UltraLite - M-Business Anywhere 编程》
- UltraLite for M-Business Anywhere: “openConnection 方法”一节 《UltraLite - M-Business Anywhere 编程》
- UltraLite.NET: “连接到数据库”一节 《UltraLite - .NET 编程》
- UltraLite.NET: “Open 方法”一节 《UltraLite - .NET 编程》



## UltraLite DBN 连接参数

当应用程序连接到多个数据库时，以名称区分数据库。

### 语法

**DBN**=*db-name*

### 缺省值

- **Palm OS** 创建者 ID。
- **所有其它平台** 文件名不带路径和扩展名（如果它们存在）。长度不能超过 16 个字符。

### 注释

UltraLite 在打开数据库以后设置数据库名。这样，客户端应用程序可以通过其名称而不是通过其文件连接到此数据库。

参数值中的所有前导空格和尾随空格均被忽略。此连接参数的值不能包含前导单引号、前导双引号或分号。

### 另请参见

- “使用连接字符串打开 UltraLite 连接” 一节第 42 页
- “UltraLite DBF 连接参数” 一节第 212 页
- UltraLite for C/C++: “连接到数据库” 一节 《UltraLite - C 及 C++ 编程》
- UltraLite for C/C++: “OpenConnection 函数” 一节 《UltraLite - C 及 C++ 编程》
- UltraLite for embedded SQL: “连接到数据库” 一节 《UltraLite - C 及 C++ 编程》
- UltraLite for M-Business Anywhere: “连接到 UltraLite 数据库” 一节 《UltraLite - M-Business Anywhere 编程》
- UltraLite for M-Business Anywhere: “openConnection 方法” 一节 《UltraLite - M-Business Anywhere 编程》
- UltraLite.NET: “连接到数据库” 一节 《UltraLite - .NET 编程》
- UltraLite.NET: “Open 方法” 一节 《UltraLite - .NET 编程》

### 示例

对连接到正在运行的、名为 Kitchener 的 UltraLite 数据库使用以下参数：

```
DBN=Kitchener;DBF=cities.udb
```

## UltraLite MIRROR\_FILE 连接参数

指定将所有数据库写入操作发送到其中的数据库镜像文件的名称（同时还会将这些写入操作发送到主数据库文件）。

### 语法

```
MIRROR_FILE=path\mirrorfile-db
```

### Default

无。

### 注释

UltraLite 提供基本的数据库文件镜像功能，来改进对潜在不可靠存储系统的容错能力。这一效果通过使用镜像文件来实现。所有数据库写入操作在发送到主数据库文件的同时也发送到镜像文件（写入开销因此加倍；读取开销不受影响）。如果从数据库文件中读取到某个损坏页，可通过从镜像文件中读取来恢复此损坏页。

使用基于文件的存储库的所有平台都支持镜像功能，包括 Palm VFS，但 Palm 记录数据库不支持镜像功能。

如果在启动数据库的同时指定了 **mirror\_file=** 选项，那么 UltraLite 会首先打开指定的文件并验证该文件是否与主数据库文件匹配，然后再继续操作。如果镜像文件不存在，则会在复制主文件时创建它。如果镜像文件不是数据库文件或者已损坏，则会报告错误，并且直到删除该镜像文件或指定了另一个镜像文件时数据库才能启动。如果镜像文件与数据库文件不匹配，则会生成 `SQL_E_MIRROR_FILE_MISMATCH` 并且数据库不会启动。当损坏页恢复时，将生成警告 `SQL_E_CORRUPT_PAGE_READ_RETRY`。（如果未镜像，或者镜像文件也损坏了，将生成错误 `SQL_E_DEVICE_ERROR` 并且数据库会暂停。）

要有效地防止介质故障，使用镜像文件时必须启用页校验和。（无论是否进行镜像，页校验和都使 UltraLite 可以在装载页面后立刻对页面损坏进行检测，从而避免引用损坏数据。）指定 `checksum_level` 数据库创建选项以启用校验和。如果在使用镜像文件时未启用校验和，UltraLite 将生成警告 `SQL_E_MIRROR_FILE_REQUIRES_CHECKSUMS`。请参见“[UltraLite checksum\\_level 创建参数](#)”一节第 168 页。

请注意，因为镜像文件是数据库文件的精确副本，所以可直接将其作为数据库来启动。`ulvalid` 实用程序将报告损坏页。请参见“[UltraLite 校验数据库实用程序 \(ulvalid\)](#)”一节第 265 页。

## 另请参见

- “使用连接字符串打开 UltraLite 连接” 一节第 42 页
- “在 UltraLite 连接参数中指定文件路径” 一节第 40 页
- “用于 UltraLite 管理工具的连接参数的优先级” 一节第 43 页
- UltraLite for C/C++: “连接到数据库” 一节 《UltraLite - C 及 C++ 编程》
- UltraLite for C/C++: “OpenConnection 函数” 一节 《UltraLite - C 及 C++ 编程》
- UltraLite for embedded SQL: “连接到数据库” 一节 《UltraLite - C 及 C++ 编程》
- UltraLite for M-Business Anywhere: “连接到 UltraLite 数据库” 一节 《UltraLite - M-Business Anywhere 编程》
- UltraLite for M-Business Anywhere: “openConnection 方法” 一节 《UltraLite - M-Business Anywhere 编程》
- UltraLite.NET: “连接到数据库” 一节 《UltraLite - .NET 编程》
- UltraLite.NET: “Open 方法” 一节 《UltraLite - .NET 编程》

## 示例

以下示例将创建一个新的连接并创建一个镜像文件:

```
Connection = DatabaseMgr.OpenConnection("DBF=c:\Dbfile.udb;  
UID=JDoe;PWD=ULdb;  
MIRROR_FILE=c:\test\MyMirrorDB.udb")
```

## UltraLite NT\_FILE 连接参数

在创建数据库时命名新的数据库文件。打开到现有数据库的连接时，此参数标识该数据库。

### 语法

```
NT_FILE=path\nt-db
```

### 缺省值

- DBF 连接参数。
- 如果 DBF 连接参数或此连接参数的值均未设置，则缺省值为 `\UltraLiteDBtore.udb`。

### 注释

应为使用同一连接字符串的 UltraLite 客户端应用程序使用 NT\_FILE 连接参数，以连接到桌面操作系统数据库以及其它平台上的数据库。

此连接参数优先于 DBF 参数。如果要从 UltraLite 管理工具进行连接，或者连接对象只连接到桌面操作系统数据库，请使用 DBF 连接参数。

NT\_FILE 的值必须满足 Windows 桌面操作系统平台的文件名要求。

路径可以是绝对路径或相对路径。如果将目录作为文件名的一部分包含于其中，则在为此文件设置路径之前，所有目录必须都已存在。UltraLite 不会自动创建目录。

参数值中的所有前导空格和尾随空格均被忽略。此连接参数的值不能包含前导单引号、前导双引号或分号。

### 另请参见

- [“使用连接字符串打开 UltraLite 连接”一节第 42 页](#)
- [“在 UltraLite 连接参数中指定文件路径”一节第 40 页](#)
- [“用于 UltraLite 管理工具的连接参数的优先级”一节第 43 页](#)
- [“UltraLite DBF 连接参数”一节第 212 页](#)
- UltraLite for C/C++: [“连接到数据库”一节《UltraLite - C 及 C++ 编程》](#)
- UltraLite for C/C++: [“OpenConnection 函数”一节《UltraLite - C 及 C++ 编程》](#)
- UltraLite for embedded SQL: [“连接到数据库”一节《UltraLite - C 及 C++ 编程》](#)
- UltraLite for M-Business Anywhere: [“连接到 UltraLite 数据库”一节《UltraLite - M-Business Anywhere 编程》](#)
- UltraLite for M-Business Anywhere: [“openConnection 方法”一节《UltraLite - M-Business Anywhere 编程》](#)
- UltraLite.NET: [“连接到数据库”一节《UltraLite - .NET 编程》](#)
- UltraLite.NET: [“Open 方法”一节《UltraLite - .NET 编程》](#)

### 示例

以下示例创建一个新连接，并为桌面操作系统平台、Palm OS 平台和 Windows Mobile 平台标识不同的数据库文件：

```
Connection = DatabaseMgr.OpenConnection("UID=JDoe;PWD=ULdb;  
NT_FILE=c:\test\MyTestDB.udb;CE_FILE=\database  
\MyCEDB.udb;PALM_FILE=MyPalmDB_MyCreatorID")
```

## UltraLite ORDERED\_TABLE\_SCAN 连接参数 [不建议使用]

该参数控制查询是通过主键索引访问行，还是直接从数据库页访问。

### 语法

**ORDERED\_TABLE\_SCAN**={ yes | no }

### 缺省值

无（直接扫描数据库页）

### 注释

如果希望以主键索引的行顺序返回结果，则重写查询，使其首先包含 ORDER BY 子句。否则，将无法利用允许此连接进行直接页扫描所带来的性能优势。只有在无法使用 ORDER BY 子句重写查询时，才使用此连接参数。

在 UltraLite 10.0.0 及之前的版本中，如果 UltraLite 优化程序未选择其它索引，则 UltraLite 使用主键索引返回结果。

### 另请参见

- “使用直接页扫描”一节第 108 页
- UltraLite for C/C++: “连接到数据库”一节 《UltraLite - C 及 C++ 编程》
- UltraLite for C/C++: “OpenConnection 函数”一节 《UltraLite - C 及 C++ 编程》
- UltraLite for embedded SQL: “连接到数据库”一节 《UltraLite - C 及 C++ 编程》
- UltraLite for M-Business Anywhere: “连接到 UltraLite 数据库”一节 《UltraLite - M-Business Anywhere 编程》
- UltraLite for M-Business Anywhere: “openConnection 方法”一节 《UltraLite - M-Business Anywhere 编程》
- UltraLite.NET: “连接到数据库”一节 《UltraLite - .NET 编程》
- UltraLite.NET: “Open 方法”一节 《UltraLite - .NET 编程》

## UltraLite PALM\_ALLOW\_BACKUP 连接参数

控制 HotSync 的备份行为，这在 UltraLite 中缺省设置为禁用。

### 语法

```
PALM_ALLOW_BACKUP={ yes | no }
```

### Default

no

### 注释

使用 UltraLite HotSync 管道对 Palm OS 设备进行的桌面操作系统备份支持此连接参数。

在 Palm 上，可使用 HotSync 将数据库备份到桌面操作系统。在大多数 UltraLite 客户端应用程序中，数据通过同步进行备份。无需使用到桌面操作系统的非正式备份。因此，UltraLite 运行库禁用 Palm 备份行为。但是，如果在进行同步的同时，部署明确需要 HotSync 将 UltraLite 数据库备份到桌面操作系统，请使用此连接参数替换 UltraLite 的缺省值。

启用 HotSync 备份之后，就不需要在 ULDBUtil 中配置备份。每次尝试同步时都会进行备份，直到将它们禁用。

### 另请参见

- “使用连接字符串打开 UltraLite 连接”一节第 42 页
- “Palm OS 的 UltraLite 数据管理实用程序 (ULDBUtil)”一节第 238 页
- UltraLite for C/C++: “连接到数据库”一节 《UltraLite - C 及 C++ 编程》
- UltraLite for C/C++: “OpenConnection 函数”一节 《UltraLite - C 及 C++ 编程》
- UltraLite for embedded SQL: “连接到数据库”一节 《UltraLite - C 及 C++ 编程》
- UltraLite for M-Business Anywhere: “连接到 UltraLite 数据库”一节 《UltraLite - M-Business Anywhere 编程》
- UltraLite for M-Business Anywhere: “openConnection 方法”一节 《UltraLite - M-Business Anywhere 编程》
- UltraLite.NET: “连接到数据库”一节 《UltraLite - .NET 编程》
- UltraLite.NET: “Open 方法”一节 《UltraLite - .NET 编程》

## UltraLite PALM\_FILE 连接参数

在创建数据库时命名新的数据库文件。打开与现有数据库的连接时，此参数标识该数据库。

### 语法 1: 基于记录的存储

**PALM\_FILE**=*name*

### 语法 2: 基于文件的存储

**PALM\_FILE**=*vfs*:[ *volume-label*:| *volume-ordinal*:]*filename*

### Default

- DBF 连接参数。
- 如果 DBF 连接参数或此连接参数的值均未设置，则缺省值为 *ulstore.udb*。

### 注释

应为使用同一连接字符串的 UltraLite 客户端应用程序使用 PALM\_FILE 连接参数，以连接到 Palm 设备以及其它平台。

此连接参数优先于 DBF 连接参数。如果要使用 UltraLite 管理工具连接，或者连接对象只连接到 Palm OS 数据库，请使用 DBF。

PALM\_FILE 的值必须满足 Palm OS 平台的文件名要求。

您不能使用 Palm 安装工具将数据库部署到 VFS 卷。而是，您需要将数据库直接复制到带有读卡器的介质或某些其它工具上。

如果在创建文件，则必须使用 DBF 连接参数，并使用 *.pdb* 扩展名命名数据库。在桌面操作系统上管理 Palm OS 数据库时使用此数据库名称。管理工具和实用程序可以使用 DBF 连接参数连接到数据库，并且数据库具有 *.pdb* 扩展名。一旦将此文件部署到设备，扩展名 *.pdb* 将被删除。随后您的应用程序必须使用 PALM\_FILE 连接参数连接到数据库，且数据库没有扩展名。

例如，如果您在管理工具桌面操作系统上使用 **DBF=CustDB.pdb** 命名文件 *CustDB.pdb*。当数据库部署到设备时，应用程序使用 **PALM\_FILE=CustDB** 连接到同一数据库。

对于 VFS 存储，请始终指定绝对文件路径。使用完整的路径名指定目录时，如果这些目录不存在，则将创建它们。

参数值中的所有前导空格和尾随空格均被忽略。此连接参数的值不能包含前导单引号、前导双引号或分号。

### 另请参见

- “使用连接字符串打开 UltraLite 连接” 一节第 42 页
- “在 UltraLite 连接参数中指定文件路径” 一节第 40 页
- “用于 UltraLite 管理工具的连接参数的优先级” 一节第 43 页
- “注册 Palm 创建者 ID” 一节 《UltraLite - C 及 C++ 编程》
- “UltraLite DBF 连接参数” 一节第 212 页
- UltraLite for C/C++: “连接到数据库” 一节 《UltraLite - C 及 C++ 编程》
- UltraLite for C/C++: “OpenConnection 函数” 一节 《UltraLite - C 及 C++ 编程》
- UltraLite for embedded SQL: “连接到数据库” 一节 《UltraLite - C 及 C++ 编程》
- UltraLite for M-Business Anywhere: “连接到 UltraLite 数据库” 一节 《UltraLite - M-Business Anywhere 编程》
- UltraLite for M-Business Anywhere: “openConnection 方法” 一节 《UltraLite - M-Business Anywhere 编程》
- UltraLite.NET: “连接到数据库” 一节 《UltraLite - .NET 编程》
- UltraLite.NET: “Open 方法” 一节 《UltraLite - .NET 编程》

### 示例

以下示例创建一个新连接，并为桌面操作系统平台、Palm OS 平台和 Windows Mobile 平台标识不同的数据库文件：

```
Connection = DatabaseMgr.OpenConnection("UID=JDoe;PWD=ULdb;  
NT_FILE=c:\test\MyTestDB.udb;CE_FILE=\database  
\MyCEDB.udb;PALM_FILE=MyPalmDB_MyCreatorID")
```



## UltraLite PWD 连接参数

定义某个用户 ID 用于验证的口令。

### 语法

**PWD**=*password*

### 缺省值

如果未设置 UID 以及 PWD，则 UltraLite 使用 **UID=DBA** 和 **PWD=sql** 打开连接。

### 注释

可将口令设置为 NULL 或空字符串，但其最大长度不可以超过 31 个字符。

数据库的每个用户都有口令。UltraLite 可支持多达四组的用户 ID/口令组合。

此连接参数未加密。但是，UltraLite 在保存口令前对其执行散列，所以只能从 Sybase Central 中修改口令。

### 另请参见

- “使用连接字符串打开 UltraLite 连接” 一节第 42 页
- “UltraLite 用户验证” 一节第 44 页
- “解释用户 ID 和口令组合” 一节第 44 页
- “UltraLite UID 连接参数” 一节第 226 页
- UltraLite for C/C++: “验证用户” 一节 《UltraLite - C 及 C++ 编程》
- UltraLite for C/C++: “连接到数据库” 一节 《UltraLite - C 及 C++ 编程》
- UltraLite for C/C++: “OpenConnection 函数” 一节 《UltraLite - C 及 C++ 编程》
- UltraLite for embedded SQL: “验证用户” 一节 《UltraLite - C 及 C++ 编程》
- UltraLite for embedded SQL: “连接到数据库” 一节 《UltraLite - C 及 C++ 编程》
- UltraLite for M-Business Anywhere: “验证用户” 一节 《UltraLite - M-Business Anywhere 编程》
- UltraLite for M-Business Anywhere: “连接到 UltraLite 数据库” 一节 《UltraLite - M-Business Anywhere 编程》
- UltraLite for M-Business Anywhere: “openConnection 方法” 一节 《UltraLite - M-Business Anywhere 编程》
- UltraLite.NET: “验证用户” 一节 《UltraLite - .NET 编程》
- UltraLite.NET: “连接到数据库” 一节 《UltraLite - .NET 编程》
- UltraLite.NET: “Open 方法” 一节 《UltraLite - .NET 编程》

### 示例

以下部分连接字符串提供用户 ID DBA 和口令 sql:

```
"UID=DBA;PWD=sql"
```

以下部分连接字符串提供用户 ID DBA 和空口令:

```
"UID=DBA;PWD=''"
```

## UltraLite RESERVE\_SIZE 连接参数

预先分配 UltraLite 数据库所需的文件系统空间，而不实际插入任何数据。保留文件系统空间意味着空间不能被其它文件用光。

### 语法

```
RESERVE_SIZE= number{ k | m | g }
```

### 缺省值

0（无保留大小）

### 注释

提供的值可以是 0 到最大数据库大小之间的任何值。分别使用 k、m 或 g 将单位指定为千字节、兆字节或千兆字节。如果未指定单位，则缺省情况下采用字节为单位。

应该使用测试数据运行数据库并观察数据库的大小，然后选择适合您的 UltraLite 部署的保留大小。

如果 RESERVE\_SIZE 的值小于数据库的大小，UltraLite 将忽略该参数。

保留文件系统空间可略微改善性能，因为它可以：

- 减少文件碎片碎裂的程度（相比数据库的逐渐增长）。
- 防止出现存储不足的内存故障。

由于 UltraLite 数据库是由数据和元数据组成的，因此数据库大小仅在被要求时（当应用程序更新数据库时）增长。

### 另请参见

- [“使用连接字符串打开 UltraLite 连接”一节第 42 页](#)
- [“UltraLite CACHE\\_SIZE 连接参数”一节第 208 页](#)
- [“UltraLite page\\_size 创建参数”一节第 182 页](#)
- UltraLite for C/C++: [“连接到数据库”一节《UltraLite - C 及 C++ 编程》](#)
- UltraLite for C/C++: [“OpenConnection 函数”一节《UltraLite - C 及 C++ 编程》](#)
- UltraLite for embedded SQL: [“连接到数据库”一节《UltraLite - C 及 C++ 编程》](#)
- UltraLite for M-Business Anywhere: [“连接到 UltraLite 数据库”一节《UltraLite - M-Business Anywhere 编程》](#)
- UltraLite for M-Business Anywhere: [“openConnection 方法”一节《UltraLite - M-Business Anywhere 编程》](#)
- UltraLite.NET: [“连接到数据库”一节《UltraLite - .NET 编程》](#)
- UltraLite.NET: [“Open 方法”一节《UltraLite - .NET 编程》](#)

### 示例

以下连接字符串片段将保留大小设置为 128 KB，这样，系统在启动时即为数据库保留此大小的系统空间。

```
"RESERVE_SIZE=128K"
```

## UltraLite START 连接参数

启动可执行 UltraLite 引擎。

### 语法

**START=***path\uleng11.exe*

### 注释

仅当要连接到当前未在运行的引擎时，才提供 StartLine (START) 连接参数。

带空格的路径需要使用引号。否则，客户端运行 `SQLE_UNABLE_TO_CONNECT_OR_START`。

### 另请参见

- “UltraLite 引擎实用程序 (uleng11)” 一节第 240 页
- “选择 UltraLite 数据管理组件” 一节第 19 页
- UltraLite for C/C++: “连接到数据库” 一节 《UltraLite - C 及 C++ 编程》
- UltraLite for C/C++: “OpenConnection 函数” 一节 《UltraLite - C 及 C++ 编程》
- UltraLite for embedded SQL: “连接到数据库” 一节 《UltraLite - C 及 C++ 编程》
- UltraLite for M-Business Anywhere: “连接到 UltraLite 数据库” 一节 《UltraLite - M-Business Anywhere 编程》
- UltraLite for M-Business Anywhere: “openConnection 方法” 一节 《UltraLite - M-Business Anywhere 编程》
- UltraLite.NET: “连接到数据库” 一节 《UltraLite - .NET 编程》
- UltraLite.NET: “Open 方法” 一节 《UltraLite - .NET 编程》

### 示例

以下命令启动位于 [程序文件] 目录中的 UltraLite 引擎:

```
Start="Program Files\uleng11.exe"
```

定义此路径的另一个方法是将整个字符串放入单引号中:

```
Start='"\Program Files\uleng11.exe"'
```

## UltraLite UID 连接参数

指定连接到数据库时使用的用户 ID。值必须为数据库的已验证用户。

### 语法

**UID**=*user*

### 缺省值

如果未设置 UID 以及 PWD，则 UltraLite 使用 **UID=DBA** 和 **PWD=sql** 打开连接。

### 注释

数据库的每个用户拥有一个用户 ID。UltraLite 可支持多达四组的用户 ID/口令组合。

UltraLite 用户 ID 不同于 MobiLink 用户名及其它 SQL Anywhere 用户 ID。用户 ID 一经创建，即不能对其进行更改。而是必须删除此用户 ID，然后添加一个新用户 ID。

不能将 UID 设置为 NULL 或空字符串。用户 ID 的最大长度为 31 个字符。用户 ID 不区分大小写。

参数值中的所有前导空格和尾随空格均被忽略。此连接参数的值不能包含前导单引号、前导双引号或分号。

### 另请参见

- “使用连接字符串打开 UltraLite 连接” 一节第 42 页
- “UltraLite 用户验证” 一节第 44 页
- “解释用户 ID 和口令组合” 一节第 44 页
- UltraLite for C/C++: “验证用户” 一节 《UltraLite - C 及 C++ 编程》
- UltraLite for C/C++: “连接到数据库” 一节 《UltraLite - C 及 C++ 编程》
- UltraLite for C/C++: “OpenConnection 函数” 一节 《UltraLite - C 及 C++ 编程》
- UltraLite for embedded SQL: “验证用户” 一节 《UltraLite - C 及 C++ 编程》
- UltraLite for embedded SQL: “连接到数据库” 一节 《UltraLite - C 及 C++ 编程》
- UltraLite for M-Business Anywhere: “验证用户” 一节 《UltraLite - M-Business Anywhere 编程》
- UltraLite for M-Business Anywhere: “连接到 UltraLite 数据库” 一节 《UltraLite - M-Business Anywhere 编程》
- UltraLite for M-Business Anywhere: “openConnection 方法” 一节 《UltraLite - M-Business Anywhere 编程》
- UltraLite.NET: “验证用户” 一节 《UltraLite - .NET 编程》
- UltraLite.NET: “连接到数据库” 一节 《UltraLite - .NET 编程》
- UltraLite.NET: “Open 方法” 一节 《UltraLite - .NET 编程》

### 示例

以下连接字符串片段为数据库提供用户 ID DBA 和口令 sql:

```
"UID=DBA;PWD=sql"
```

---

# UltraLite 实用程序

## 目录

支持的退出代码 .....	228
UltraLite 的 Interactive SQL 实用程序 (dbisql) .....	229
UltraLite 实用程序的 SQL 预处理器 (sqlpp) .....	232
UltraLite 创建数据库实用程序 (ulcreate) .....	235
Palm OS 的 UltraLite 数据管理实用程序 (ULDBUtil) .....	238
UltraLite 引擎实用程序 (uleng11) .....	240
UltraLite 引擎停止实用程序 (ulstop) .....	241
UltraLite 消除数据库 (ulerase) .....	242
Palm OS 的 UltraLite HotSync 管道安装实用程序 (ulcond11) .....	243
UltraLite 信息实用程序 (ulinfo) .....	245
UltraLite 初始化数据库实用程序 (ulinit) .....	249
UltraLite 装载 XML 到数据库实用程序 (ulload) .....	252
UltraLite 同步实用程序 (ulsync) .....	255
同步配置文件选项 .....	258
UltraLite 卸载旧数据库实用程序 (ulunload) .....	260
UltraLite 卸载旧数据库实用程序 (ulunloadold) .....	263
UltraLite 校验数据库实用程序 (ulvalid) .....	265

---

UltraLite 包含一组实用程序，设计用于在命令提示符下执行基本数据库管理活动。许多 UltraLite 实用程序的功能与 SQL Anywhere 实用程序的功能类似。但是，使用选项的方法会有所不同。对于这些选项的 UltraLite 实现，请始终查阅 UltraLite 参考文档。

### **注意**

除非另行说明，本章中介绍的实用程序的选项是区分大小写的。请完全按照所显示的样子键入选项。

## 支持的退出代码

Ulcreate、uload、ulsync 和 ulunload 实用程序返回退出代码，用于指示实用程序试图完成的操作成功与否。0 指示操作成功。任何其它值都表示操作失败。

退出代码	状态	说明
0	EXIT_OKAY	操作成功。
1	EXIT_FAIL	操作失败。
3	EXIT_FILE_ERROR	无法找到数据库。
4	EXIT_OUT_OF_MEMORY	已用尽设备的动态内存。
6	EXIT_COMMUNICATIONS_FAIL	与 UltraLite 引擎对话时，产生通信错误。
9	EXIT_UNABLE_TO_CONNECT	提供了无效的 UID 或 PWD，因此不能连接到数据库。
12	EXIT_BAD_ENCRYPT_KEY	加密密钥遗失或无效。
13	EXIT_DB_VER_NEWER	检测到该数据库的版本是不兼容的。必须将数据库升级到较新的版本。
255	EXIT_USAGE	无效的命令行选项。

## UltraLite 的 Interactive SQL 实用程序 (dbisql)

执行 SQL 命令或运行命令文件。

### 语法

**dbisql -c "connection-string" -ul [ options ] [ dbisql-command | command-file ]**

选项	说明
<b>@data</b>	<p>从指定的环境变量或配置文件中读取选项。如果存在具有相同名称的环境变量和配置文件，则使用环境变量。</p> <p>如果要保护口令或配置文件中的其它信息，可以使用文件隐藏实用程序对配置文件的内容进行模糊处理。请参见“使用配置文件”一节《SQL Anywhere 服务器 - 数据库管理》。</p>
<b>-c "connection-string"</b>	<p>必需。连接到 DBF 或 <i>connection-string</i> 的 <i>file_name</i> 参数所标识的数据库。如果未指定用户 ID 和口令，则会假定缺省 UID 为 <b>DBA</b> 且 PWD 为 <b>sql</b>。</p>
<b>-d delimiter</b>	<p>指定命令分隔符。分隔符两侧的引号是可选的，但是，当分隔符对于命令解释程序自身来说具有特殊含义时，引号则是必需的。</p> <p>命令分隔符用于该 Interactive SQL 会话中的所有连接，无论存储在数据库中的设置是什么。</p>
<b>-dl</b>	<p>将用户显式执行的所有语句都回写到命令窗口 (STDOUT)。在调试 SQL 脚本或在 Interactive SQL 处理长 SQL 脚本时，这样可以提供有用的反馈。</p>
<b>-f filename</b>	<p>打开（但不运行）名为 <i>filename</i> 的文件。如果文件名包含空格，请将文件名用引号括起来。</p> <p>如果有 -f 选项，将忽略 -c 选项；即不会连接数据库。</p> <p>如果未包含完整的文件路径，则假定一个与当前目录相关的路径。如果文件不存在，或它实际上是目录而非文件，则 Interactive SQL 会输出错误消息，然后关闭。</p> <p>只有在将 Interactive SQL 作为窗口式应用程序运行时，才支持此选项。</p>
<b>-nogui</b>	<p>在命令提示符模式下运行。如果指定 <i>dbisql-command</i> 或 <i>command-file</i>，则假定已指定 -nogui。</p>

选项	说明
<b>-onerror behavior</b>	<p>控制在从指定的命令文件中读取数据期间遇到错误时所采取的操作。定义以下支持的 <i>behavior</i> 值之一：</p> <ul style="list-style-type: none"> <li>● <b>Stop</b> Interactive SQL 停止执行语句。</li> <li>● <b>Prompt</b> Interactive SQL 提示用户查看是否要继续。</li> <li>● <b>Continue</b> 忽略错误，Interactive SQL 继续执行语句。</li> <li>● <b>Exit</b> Interactive SQL 终止。</li> <li>● <b>Notify_Continue</b> 报告错误，并提示用户按 Enter 键或单击 [确定] 继续。</li> <li>● <b>Notify_Stop</b> 报告错误，并提示用户按 Enter 键或单击 [确定] 停止执行语句。</li> <li>● <b>Notify_Exit</b> 报告错误，并提示用户按 Enter 键或单击 [确定] 终止 Interactive SQL。</li> </ul>
<b>-q</b>	将实用程序设置为在安静模式下运行。取消信息性标题、版本号和状态消息。但是，仍然会显示错误消息。
<b>-ul</b>	<p>缺省情况下，连接到 UltraLite 数据库。</p> <p>缺省情况下，Interactive SQL 会假定您连接的是 SQL Anywhere 数据库。如果指定 <b>-ul</b> 选项，缺省设置会改变为 UltraLite 连接。无论数据库设置的缺省值类型如何，您都可以连接到 SQL Anywhere 或 UltraLite 数据库。通过从 <a href="#">连接</a> 窗口的下拉列表中选择数据库类型来执行此操作。</p>
<b>-version</b>	显示 Interactive SQL 的版本号。您也可以在 Interactive SQL 中查看版本号；在 [帮助] 菜单中，选择 [关于 Interactive SQL]。
<b>-x</b>	扫描命令，但不执行这些命令。对于检查长命令文件的语法错误，此选项很有用。
<b>SQL-command command-file</b>	<p>执行 SQL 语句或指定的 <i>command-file</i>。</p> <p>如果未指定 <i>SQL-statement</i> 或 <i>command-file</i>，则 Interactive SQL 将进入交互模式，在这种模式下，可以在命令窗口中键入命令。</p>

## 注释

Interactive SQL 允许您执行 SQL 命令或运行命令文件。还提供关于以下内容的反馈：

- 受影响的行数
- 各项命令所需的时间
- 查询的执行计划
- 任何错误消息



当连接到 UltraLite 数据库时，SQL Anywhere 特有的菜单项将不会在界面中显示。例如，[工具] » [查寻过程名] 或 [工具] » [索引顾问]。

在 UltraLite 中，归类包括一个代码页加上一个排序顺序。因此，代码页的编号与显示为 UltraLite 归类名一部分的编号相对应。要查看支持的归类（及其相应代码页）的列表，在命令提示符下运行 `ulcreate -l`。

可使用 INPUT、OUTPUT 或 READ 语句的 ENCODING 子句，指定在读取或写入文件时要使用的代码页。例如，在英文 32 位 Windows 台式计算机上，窗口式程序使用 1252 (ANSI) 代码页。如果要让 Interactive SQL 读取用 297 (IBM France) 代码页创建的名为 `status.txt` 的文件，可使用以下语句：

```
READ
ENCODING 297
status.txt;
```

退出代码是零（成功）或非零（失败）。只有在批处理模式中运行 Interactive SQL（即执行包含 SQL 语句或脚本文件名称的命令）时才会设置非零退出代码。

在命令行模式下，Interactive SQL 将设置程序退出代码以指示成功或失败。在 Windows 操作系统中，环境变量 `ERRORLEVEL` 被设置为程序退出代码。

### 另请参见

- “使用配置文件”一节 《SQL Anywhere 服务器 - 数据库管理》
- “文件隐藏实用程序 (dbfhide)”一节 《SQL Anywhere 服务器 - 数据库管理》
- “INPUT 语句 [Interactive SQL]”一节 《SQL Anywhere 服务器 - SQL 参考》
- “OUTPUT 语句 [Interactive SQL]”一节 《SQL Anywhere 服务器 - SQL 参考》
- “READ 语句 [Interactive SQL]”一节 《SQL Anywhere 服务器 - SQL 参考》
- “UltraLite 连接参数”第 207 页
- “UltraLite 创建数据库实用程序 (ulcreate)”一节第 235 页
- “支持的退出代码”一节第 228 页

### 示例

以下命令针对 UltraLite 的 `CustDB.udb` 数据库运行命令文件 `mycom.sql`。因为未定义用户 ID 和口令，所以假定缺省用户 ID 为 **DBA** 且口令为 **sql**。如果命令文件中存在错误，则该过程将终止。

```
dbisql -ul -c DBF=CustDB.udb -onerror exit mycom.sql
```

## UltraLite 实用程序的 SQL 预处理器 (sqlpp)

预处理包含嵌入式 SQL (ESQL) 的 C/C++ 程序，这样，可以在运行编译器之前生成该程序所需的代码。请注意，为保持完整性，下表描述了整个选项集，但是，UltraLite 的相关选项只有 **-eu** 和 **-wu**。

### 语法

**sqlpp -u** [ *options* ] *esql-filename* [ *output-filename* ]

选项	说明
<b>-d</b>	生成减小数据空间大小但增大代码大小的代码。数据结构在使用之前执行时会得到重用和初始化。
<b>-e flag</b>	<p>此选项将任何未包含在指定标准中的静态嵌入式 SQL 标记为错误。<i>level</i> 值表示要使用的标准。例如，<code>sqlpp -e c03 ...</code> 标记任何未包含在核心 SQL/2003 标准中的语法。</p> <p><i>level</i> 的允许值有：</p> <ul style="list-style-type: none"> <li>● <b>c03</b> 标记不是核心 SQL/2003 语法的语法</li> <li>● <b>p03</b> 标记非完整 SQL/2003 语法的语法</li> <li>● <b>c99</b> 标记不是核心 SQL/1999 语法的语法</li> <li>● <b>p99</b> 标记非完整 SQL/1999 语法的语法</li> <li>● <b>e92</b> 标记非入门级 SQL/1992 语法的语法</li> <li>● <b>i92</b> 标记非中级 SQL/1992 语法的语法</li> <li>● <b>f92</b> 标记非完整 SQL/1992 语法的语法</li> <li>● <b>t</b> 标记非标准主机变量类型</li> <li>● <b>u</b> 标记 UltraLite 不支持的语法</li> </ul> <p>为了与以前的 SQL Anywhere 版本兼容，也可指定 <b>e</b>、<b>I</b> 和 <b>f</b>（分别对应 <b>e92</b>、<b>i92</b> 和 <b>f92</b>）。</p>
<b>-h width</b>	在 <i>.c</i> 文件中，将 <code>sqlpp</code> 输出的拆分行的最大长度限制为 <i>width</i> 。在拆分行的末尾添加反斜线字符，这样 C 编译器就可以将这些拆分行作为一个连续的行进行分析。缺省值为无最大行长度（缺省情况下不拆分输出行）。
<b>-k</b>	通知预处理器要编译的程序包括 <code>SQLCODE</code> 的用户声明。
<b>-n</b>	<p>在生成的代码的相应位置上使用 <code>#line</code> 指令，可在 C 文件中生成行号信息。</p> <p>此选项用于报告源代码错误，以及调试 <i>esql-filename</i> 文件（而不是 <i>output-filename</i> 文件）中这些行号处的源代码。</p>

选项	说明
<b>-o</b> <i>O/S spec</i>	不适用于 UltraLite。
<b>-q</b>	将实用程序设置为在安静模式下运行。取消信息性标题、版本号和状态消息。但是，仍然会显示错误消息。
<b>-r</b>	不适用于 UltraLite。
<b>-s</b> <i>string-length</i>	设置预处理器要放入 C 文件的最大大小字符串。对于长度大于此值的字符串，将使用一组字符（'a'、'b'、'c' 等）对其进行初始化。大多数 C 编译器都对可以处理的字符串大小有限制。此选项用于设置其上限。缺省值是 500。
<b>-u</b>	UltraLite 必需的。生成 UltraLite 数据库特有的必需输出。
<b>-w</b> <i>level</i>	<p>将不符合要求的 SQL 语法标记为警告。<i>level</i> 值表示要使用的标准。例如，<code>sqlpp -w c03 ...</code> 标记任何未包含在核心 SQL/2003 语法中的 SQL 语法。</p> <p><i>level</i> 的允许值有：</p> <ul style="list-style-type: none"> <li>● <b>c03</b> 标记不是核心 SQL/2003 语法的语法</li> <li>● <b>p03</b> 标记非完整 SQL/2003 语法的语法</li> <li>● <b>c99</b> 标记不是核心 SQL/1999 语法的语法</li> <li>● <b>p99</b> 标记非完整 SQL/1999 语法的语法</li> <li>● <b>e92</b> 标记非入门级 SQL/1992 语法的语法</li> <li>● <b>i92</b> 标记非中级 SQL/1992 语法的语法</li> <li>● <b>f92</b> 标记非完整 SQL/1992 语法的语法</li> <li>● <b>t</b> 标记非标准主机变量类型</li> <li>● <b>u</b> 标记 UltraLite 不支持的语法</li> </ul> <p>为了与以前的 SQL Anywhere 版本兼容，也可指定 <b>e</b>、<b>I</b> 和 <b>f</b>（分别对应 <b>e92</b>、<b>i92</b> 和 <b>f92</b>）。</p>
<b>-x</b>	将多字节字符串更改为转义序列，以便它们可以通过编译器。
<b>-z</b> <i>collation-sequence</i>	指定归类序列。

### 注释

此预处理器将 *input-file* 中的 SQL 语句转换为 C/C++，并将结果写入 *output-filename*。含有嵌入式 SQL 的源文件的常规扩展名是 *sql*。缺省的 *output-filename* 为基本名 *esql-filename* 加上扩展名 *c*。但是，如果 *esql-filename* 已经有 *.c* 扩展名，则缺省的输出扩展名是 *.cc*。

归类序列用于帮助预处理器理解程序的源代码所使用的字符。例如，标识适用于标识符的字母字符。在 UltraLite 中，归类包括一个代码页加上一个排序顺序。如果没有指定 `-z`，则预处理器会尝试根据操作系统确定要使用的合理归类。

要查看支持的归类（及其相应代码页）的列表，在命令提示符下运行 `ulcreate -l`。

### 提示

SQL 预处理器 (sqlpp) 可在编译时在嵌入式 SQL 应用程序中标记静态 SQL 语句。此特性在开发 UltraLite 应用程序时对于验证 SQL 语句的 UltraLite 兼容性特别有用。使用 `-e` 和/或 `-w` 选项，可以测试 SQL Anywhere 与 UltraLite 应用程序的 SQL 兼容性。有关 SQL Flagger 的概述，请参见“使用 SQL Flagger 测试 SQL 遵从性”一节《SQL Anywhere 服务器 - SQL 的用法》。

### 另请参见

- “SQL Anywhere 嵌入式 SQL” 《SQL Anywhere 服务器 - 编程》
- “UltraLite 字符集” 一节第 32 页

### 示例

以下命令在安静模式下为 UltraLite 应用程序预处理 `srcfile.sqc` 嵌入式 SQL 文件。

```
sqlpp -u -q MyEsqFile.sqc
```

## UltraLite 创建数据库实用程序 (ulcreate)

用您定义的属性创建 UltraLite 数据库。

### 语法

**ulcreate** [ *options* ] [ *new-database-file* ]

选项	说明
<b>-c</b> " <i>connection-string</i> "	创建 DBF 或 <i>connection-string</i> 的 <i>file_name</i> 参数所标识的数据库。如果未指定用户 ID 和口令，则会假定缺省 UID 为 <b>DBA</b> 且 PWD 为 <b>sql</b> 。如果您指定了用户 ID 和口令，则需使用它们访问数据库。  如果未在连接字符串中以参数方式提供文件名，ulcreate 将在命令的末尾检查指定为 <i>new-database-file</i> 的文件。
<b>-g</b> <i>global-ID</i>	将初始数据库 ID 设置为所指派的 INTEGER 值。此初始值与具有全局自动增量列的新行的分区大小一起使用。在部署应用程序时，必须为每个数据库指派不同的标识号范围，目的是为了与 MobiLink 服务器同步。请参见“ <a href="#">UltraLite global_database_id 选项</a> ”一节第 204 页。
<b>-l</b>	列出可用的归类序列，然后退出。
<b>-o</b> [ <i>extended-options</i> ]	指定 UltraLite 数据库创建参数的以分号分隔的列表。请参见“ <a href="#">为 UltraLite 选择数据库创建参数</a> ”一节第 30 页。
<b>-ol</b>	列出可用的数据库创建参数，然后退出。请参见“ <a href="#">为 UltraLite 选择数据库创建参数</a> ”一节第 30 页。
<b>-p</b> <i>creator-ID</i>	将数据库安装到记录存储库时 Palm OS 所需要的选项。使用 UltraLite 客户端应用程序的指定四字符 <i>creator-ID</i> 创建数据库。如果您正在将数据库部署到 VFS 存储库，则无需使用此选项。
<b>-q</b>	将实用程序设置为在安静模式下运行。取消信息性标题、版本号和状态消息。但是，仍然会显示错误消息。
<b>-t</b> <i>file</i>	将包含公共受信任根证书的特定文件装载到数据库中，并且不再需要提供 <i>trusted_certificates</i> 同步参数。服务器验证需要此证书。
<b>-v</b>	打印详细消息。
<b>-y</b>	如果数据库文件存在，则将其覆盖。
<b>-z</b> <i>collation-sequence</i>	指定要使用的归类的标签。

选项	说明
<i>new-database-file</i>	创建一个具有指定名称的文件。只有在未使用连接字符串以设置初始数据库参数（如用户 ID (UID) 或口令 (PWD)）时，才使用此独立文件名。确保所设置的独立文件名适用于您的平台。

## 注释

如果未设置任何数据库属性，`ulcreate` 则会创建一个不区分大小写、归类序列取决于当前地区的数据库。

无论数据库是否区分大小写，数据库口令始终区分大小写。数据库是否区分大小写取决于是否使用了 "case=respect" 创建参数。

归类序列用于数据库中的所有字符串比较。在 UltraLite 中，归类包括一个代码页加上一个排序顺序。如果没有指定 `-z`，则 `ulcreate` 会尝试根据桌面操作系统的当前地区确定要使用的合理归类。

要查看支持的归类（及其相应代码页）的列表，在命令提示符下运行 `ulcreate -l`。

设备的操作系统确定是否应使用 UTF-8 编码。

写入桌面操作系统的 Palm OS 数据库必须以 `.pdb` 扩展名标识。但是，一旦将数据库部署到设备，扩展名将被删除。有关文件名格式的详细信息，请参见“[Palm OS](#)”一节第 41 页。

如果目标是 VFS 卷，则不能使用 Palm 安装工具部署 UltraLite 数据库。而是，需要使用读卡器或某些其它工具将数据库直接复制到介质上。

此实用程序会返回错误代码。除 0 之外的任何值均意味着操作失败。

## 另请参见

- [“创建和配置 UltraLite 数据库” 第 25 页](#)
- [“在 UltraLite 连接参数中指定文件路径” 一节第 40 页](#)
- [“UltraLite 连接参数” 第 207 页](#)
- [“支持的退出代码” 一节第 228 页](#)
- [“支持的归类和替代归类” 一节 《SQL Anywhere 服务器 - 数据库管理》](#)
- [“注册 Palm 创建者 ID” 一节 《UltraLite - C 及 C++ 编程》](#)
- [“UltraLite 大小写创建参数” 一节第 167 页](#)
- [“字符集编码的 UltraLite 平台要求” 一节第 33 页](#)
- [“MobiLink 模型简介” 一节 《MobiLink - 入门》](#)

## 示例

创建名为 `test.udb` 的 UltraLite 数据库，它是不区分大小写、归类序列取决于当前地区的非 Unicode 数据库：

```
ulcreate test.udb
```

创建一个区分大小写、名为 `test.udb` 的数据库，从而将其创建为具有与 ISO 兼容的日期格式和顺序的数据库：

```
ulcreate -c DBF=test.udb -o case=respect;date_format=YYYY-MM-DD;date_order=YMD
```

创建名为 *test.udb* 的加密数据库，加密密钥为 **afvc\_1835**:

```
ulcreate -c "DBF=test.udb;DBKEY=afvc_1835"
```

## Palm OS 的 UltraLite 数据管理实用程序 (ULDBUtil)

从设备中删除数据库，或在下次同步时备份数据库。

### 语法

无。

### 注释

可以使用此实用程序执行以下任务：

- 当设备在不同用户间共享时，从设备删除数据库。删除文件可以节省空间或维护保密性。然后可以重新安装数据库，甚至可以让数据库创建一个新的未填充的数据库。
- 在下一同步时，备份数据库。使用此功能执行初始同步，然后备份数据库。这样就允许您将数据库部署到其它设备上，因此它们将不需要执行初始同步。

此实用程序安装为以下文件：

```
install-dir\UltraLite\Palm\68k\ULDBUtil.prc
```

ULDBUtil 一旦安装，就使用文件名过滤器。此过滤器只检测满足 UltraLite 扩展名要求的 UltraLite 数据库文件。文件名过滤器列表排除具有以下扩展名的所有文件：*.bak*、*.dat*、*.dba*、*.htm*、*.html*、*.in*、*.jpeg*、*.jpg*、*.prc*、*.pqa* 和 *.tda*。确保未使用此过滤器所使用的任何扩展名命名 UltraLite 数据库。

### ◆ 从 Palm OS 设备删除 UltraLite 应用程序数据

1. 切换到 ULDBUtil。
2. 如果您的设备有扩展卡，则选择应用程序数据库文件将从中删除的介质（内部/外部卷或基于记录的介质）。
3. 从 UltraLite 版本 11 数据库列表选择一个数据库。
4. 在 Palm 设备上单击 [删除] 删除数据。

以下过程只适用于通过 HotSync 管道部署的数据库。

### ◆ 从 Palm OS 设备备份数据库到台式机

1. 切换到 ULDBUtil。
2. 请选择 [备份] 选项，这样 HotSync 就会知道在下一同步尝试时备份数据库。需要为每个所需的后续备份选择此选项。存储在 VFS 卷（如卡）上的数据库禁用此选项。

#### 提示

如果想启用自动备份，从而避免每次同步时都要选择此选项，请使用 PALM\_ALLOW\_BACKUP 参数。请参见“UltraLite PALM\_ALLOW\_BACKUP 连接参数”一节第 220 页。



**另请参见**

- [“Palm OS 的 UltraLite HotSync 管道安装实用程序 \(ulcond11\)”](#) 一节第 243 页

**示例**

无。

## UltraLite 引擎实用程序 (uleng11)

管理来自在 32 位 Windows 桌面操作系统和 Windows Mobile 上应用程序的并发 UltraLite 数据库连接。

### 语法

**uleng11**

### 注释

UltraLite 引擎在启动时不显示消息窗口。

链接到 UltraLite 引擎需要一组与 UltraLite 运行库所使用的库不同的库。如果还需要启用了 TLS 的同步或 AES FIPS 数据库加密，则还需要其它库。

请参见：

- [“编译和链接应用程序”一节 《UltraLite - C 及 C++ 编程》](#)
- [“通过 AES\\_FIPS 数据库加密部署 UltraLite”一节第 50 页](#)
- [“通过启用 TLS 的同步部署 UltraLite”一节第 51 页](#)

### 另请参见

- [“使用 UltraLite 数据库”第 61 页](#)
- [“选择 UltraLite 数据管理组件”一节第 19 页](#)
- [“UltraLite 引擎停止实用程序 \(ulstop\)”一节第 241 页](#)
- [“UltraLite START 连接参数”一节第 225 页](#)

## UltraLite 引擎停止实用程序 (ulstop)

在 32 位 Windows 桌面操作系统和 Windows Mobile 上，停止 UltraLite 引擎。

### 语法

**ulstop**

### 注释

在开发期间使用 **ulstop** 手工关闭引擎。通常，在活动部署中不需要 **ulstop**。

此实用程序没有选项。

### 另请参见

- [“选择 UltraLite 数据管理组件”一节第 19 页](#)
- [“UltraLite 引擎实用程序 \(uleng11\)”一节第 240 页](#)

## UltraLite 消除数据库 (ulerase)

消除 UltraLite 数据库。

### 语法

```
ulerase -c "connection-string" [options] [ dbname ]
```

选项	说明
-c "connection-string" keyword=value	必需。连接到 DBF 或 <i>connection-string</i> 的 <i>file_name</i> 参数所标识的数据库。如果未指定用户 ID 和口令，则会假定缺省 UID 为 <b>DBA</b> 且 PWD 为 <b>sql</b> 。
-ek key	指定加密数据库的加密密钥。
-ep	指定需要系统提示输入加密密钥。
-q	将实用程序设置为在安静模式下运行。取消信息性标题、版本号和状态消息。但是，仍然会显示错误消息。
-v	打印详细消息。

### 注释

数据库必须可以访问。用户 ID 和口令组合必须允许连接，否则将无法消除数据库。

加密数据库需要连接字符串提供的密钥，或者使用 -ek <key> 或 -ep 之一。

## Palm OS 的 UltraLite HotSync 管道安装实用程序 (ulcond11)

使用 HotSync 管道安装并注册每个 Palm OS 数据库，这样，该管道就可以管理每个数据库的 HotSync 同步操作。也可以使用此实用程序来卸载管道。

### 语法

**ulcond11 -c "connection-string" [ options ] creator-ID**

选项	说明
<i>creator-ID</i>	必需。设置使用管道的应用程序的创建者 ID。如果指定创建者 ID 的管道已经存在，则用新管道替换它。
<b>-a</b>	将附加的数据库连接字符串追加到使用 <b>-c</b> 选项配置的连接字符串上。用管道注册多个数据库，使用此选项。
<b>-c "connection-string sync_profile=profile_name{}</b>	必需。连接到 <i>connection-string</i> 的 DBF 参数所标识的设备上的 Palm 数据库。您定义的连接字符串通过管道注册已部署 Palm 数据库。将连接参数作为管道的部分配置信息进行存储。  <b>sync_profile</b> 指定要使用的同步配置文件以及任意合并参数。缺省配置文件为 <b>ul_palm_conduit</b> 。  如果未指定用户 ID 和口令，则会假定缺省 UID 为 <b>DBA</b> 且 PWD 为 <b>sql</b> 。
<b>-d filename</b>	设置插件 <i>.dll</i> 文件的名称。  此实用程序用一个名为 <i>PluginDLL</i> 的值创建注册表项 ( <i>Software\Sybase\SQL Anywhere\version\Conduit\creator-ID</i> )。 <i>PluginDLL</i> 的值为所设置的 <i>filename</i> 。  提供一个空字符串 ( <b>-d""</b> ) 以从注册表中清除现有文件名。
<b>-n name</b>	设置 HotSync 管理器显示的名称。缺省值为 <b>Conduit</b> 。 请不要将此选项与 <b>-u</b> 选项一起使用。
<b>-q</b>	将实用程序设置为在安静模式下运行。取消信息性标题、版本号和状态消息。但是，仍然会显示错误消息。  如果在 Window Vista 上以正确的权限运行 <b>ulcond11</b> ，在没有错误要报告时，将立即关闭窗口。否则，将应用五秒的延迟。
<b>-QQ</b>	将实用程序设置为在安静模式下运行。取消信息性标题、版本号和状态消息。但是，仍然会显示错误消息。  如果在 Window Vista 上以正确的权限运行 <b>ulcond11</b> ，将立即关闭窗口。

选项	说明
<b>-u</b>	卸载该创建者 ID 的管道。如果未指定 <b>-u</b> ，则安装管道。如果指定了 <b>-u</b> ，则不要指定 <b>-n</b> 。

### 注释

ulcond11 是命令行实用程序。在 Windows Vista 上，ulcond11 运行权限更高。可执行文件窗口不会在五秒内关闭，使您可以查看输出。

HotSync 记录每次同步发生的时间以及每个已安装的管道是否像预期的那样进行工作。HotSync 日志文件位于 Palm Desktop 安装目录的 *User* 子目录中。

### 另请参见

- [“UltraLite 连接参数” 第 207 页](#)
- [“Palm OS 的 UltraLite 数据管理实用程序 \(ULDBUtil\)” 一节第 238 页](#)
- [“部署 UltraLite HotSync 管道” 一节第 53 页](#)
- [“UltraLite 同步流的网络协议选项” 一节第 161 页](#)
- [“Palm OS 上的 HotSync” 一节第 132 页](#)
- [“注册 Palm 创建者 ID” 一节 《UltraLite - C 及 C++ 编程》](#)

### 示例

以下命令为创建者 ID 是 **Syb2** 的应用程序安装名为 **CustDB** 的管道。下面为 CustDB 示例应用程序的设置：

```
ulcond11 -c "DBF=custdb.udb;UID=DBA;PWD=sql" -n CustDB Syb2
```

以下命令为同一个 CustDB 示例应用程序卸载管道：

```
ulcond11 -u Syb2
```

以下命令使用 Palm 的默认同步配置文件为 palmdb 安装管道：

```
ulcond11 ... -c
"dbf=palmdb;sync_profile=ul_palm_conduit{stream.host=override_host}" ...
```

## UltraLite 信息实用程序 (ulinfo)

显示有关 UltraLite 数据库的信息并更改或清除 `global_id` 或 `ml_remote_id` 数据库选项。

### 语法

**ulinfo -c "connection-string" [ options ]**

选项	说明
<b>-c "connection-string"</b>	必需。连接到 DBF 或 <i>connection-string</i> 的 <code>file_name</code> 参数所标识的数据库。如果未指定用户 ID 和口令，则会假定缺省 UID 为 <b>DBA</b> 且 PWD 为 <b>sql</b> 。
<b>-g ID</b>	将初始全局数据库 ID 设置为指派的值。数据库将此值用于所有具有全局自动增量列的新行。数据库将此基值用于与每个附加行和/或列相关联的自动增量 ID。  在部署应用程序时，必须为每个数据库指派不同的标识号，目的是为了与 MobiLink 服务器同步。
<b>-oa</b>	如果过程标识数据库由 UltraLite 的前一版本创建（将导致数据库升级），则将该过程取消。
<b>-or</b>	以只读模式打开数据库。UltraLite 会创建原始文件的副本，然后可以使用此副本测试脚本而不会变更数据库。完成后会放弃对复制文件的更改。  如果直接从桌面操作系统连接到已部署到 Windows Mobile 设备的数据库，则不支持此选项。
<b>-ou</b>	如果数据库由较旧版本的 UltraLite 创建，则升级数据库。
<b>-q</b>	将实用程序设置为在安静模式下运行。取消信息性标题、版本号和状态消息。但是，仍然会显示错误消息。
<b>-r ID</b>	将初始 <code>ml_remote_id</code> 设置为指派的值。缺省情况下，新的 UltraLite 数据库将 MobiLink 远程 ID 设置为 NULL。选择以下内容可以保留此缺省值：UltraLite 和 <code>dbmsync</code> 都在启动同步时自动将 MobiLink 远程 ID 设置为唯一用户 ID (UUID)。
<b>-rc</b>	将 MobiLink 远程 ID 设置为 NULL。
<b>-v</b>	打印详细消息。除指定数据库的数据库内部信息之外，还显示当前数据库属性。

### 注释

始终会显示打开 UltraLite 数据库时生成的警告消息，除非使用了 `-q` 选项。

当使用安静标志 (`-q`) 时，`ulinfo` 实用程序显示有关现有同步配置文件的信息以及待执行 SQL 直通脚本的编号。例如：

```
Sync profile cv_prosync:
"TableOrder=cv_prosync;MobiLinkId=cv_prosync;ScriptVersion=test;"
Number of available SQL pass-through scripts: 0
```

### 另请参见

- [“UltraLite 连接参数” 第 207 页](#)
- [“UltraLite global\\_database\\_id 选项” 一节第 204 页](#)
- [“UltraLite ml\\_remote\\_id 选项” 一节第 205 页](#)

### 示例

显示已经同步的、名为 *sample.udb* 的文件的基本数据库内部信息：

```
ulinfo -c DBF=cv_dbattr.udb

ulinfo -c DBF=cv_dbattr.udb
Utility Version 11.*suppressed*
Collation: 1252LATIN1
Number of Users: 1
  1. User: 'DEA'
Page size: 4096
Default index maximum hash size: 4
Checksum level: 0
MobiLink Remote ID: not set
Global database ID: 1000
Global autoincrement usage: 0%
Number of tables: 3
Number of columns: 7
Number of publications: 3
Number of tables that will always be uploaded: 0
Number of tables that are never synchronized: 0
Number of primary Keys: 3
Number of foreign Keys: 0
Number of indexes: 0
This database has not yet been synchronized.
Synchronization by publication number:
  1. Publication cv_sync
    Number of rows in next upload: 0
    Last download: 1900-01-01 00:00:00.000
  2. Publication cv_syncPub2
    Number of rows in next upload: 0
    Last download: 1900-01-01 00:00:00.000
  3. Publication cv_syncPub3
    Number of rows in next upload: 0
    Last download: 1900-01-01 00:00:00.000
Number of available SQL pass-through scripts: 0
ulsync ...
Utility Version 11.*suppressed*
Results of this synchronization:
Succeeded
Download timestamp: 20XX-XX-XX XX:XX:XX.XXXXXX
Upload OK
No ignored rows
No part download remaining
Authentication value: 1000 (0x3e8)
ulinfo -c dbf=cv_dbattr.udb
Utility Version 11.*suppressed*
Collation: 1252LATIN1
Number of Users: 1
  1. User: 'DEA'
Page size: 4096
```



```

Default index maximum hash size: 4
Checksum level: 0
MobiLink Remote ID: XXXXXXXX-XXXX-XXXX-XXXX-XXXXXXXXXXXX
Global database ID: 1000
Global autoincrement usage: 0%
Number of tables: 3
Number of columns: 7
Number of publications: 3
Number of tables that will always be uploaded: 0
Number of tables that are never synchronized: 0
Number of primary Keys: 3
Number of foreign Keys: 0
Number of indexes: 0
Last synchronization completed successfully
Download occurred: *suppressed*
Upload OK
Upload rows not ignored
No partial downloads
Actual MobiLink Authentication value: 1000
Authentication valid
Synchronization by publication number:
  1. Publication cv_sync
     Number of rows in next upload: 0
     Last download: 20XX-XX-XX XX:XX:XX.XXXXXX
  2. Publication cv_syncPub2
     Number of rows in next upload: 0
     Last download: 1900-01-01 00:00:00.000
  3. Publication cv_syncPub3
     Number of rows in next upload: 0
     Last download: 1900-01-01 00:00:00.000
Number of available SQL pass-through scripts: 0

```

显示名为 *CustDB.udb* 的文件的数据库内部信息，并通过启用详细消息显示数据库属性：

```

ulinfo -c DBF=CustDB.udb -v

Utility Version 11.*suppressed*
Database information
Database name: custdb
Disk file 'C:\Documents and Settings\All Users\Shared Documents\SQL Anywhere
11\Samples\UltraLite\CustDB\custdb.udb'
Collation: 1252LATIN1
Number of Users: 1
  1. User: 'DBA'
Page size: 4096
Default index maximum hash size: 8
Checksum level: 0
MobiLink Remote ID: not set
Global database ID: not set
Encryption: None
Character encoding set: 1252LATIN1
Case sensitive: OFF
Date format: YYYY-MM-DD
Date order: YMD
Nearest century: 50
Numeric precision: 30
Numeric scale: 6
Time format: HH:NN:SS.SSS
Timestamp format: YYYY-MM-DD HH:NN:SS.SSS
Timestamp increment: 1
Number of tables: 6
Number of columns: 16
Number of publications: 1
Number of tables that will always be uploaded: 0

```

```
Number of tables that are never synchronized: 1
Number of primary Keys: 6
Number of foreign Keys: 2
Number of indexes: 3
This database has not yet been synchronized.
Synchronization by publication number:
  1. Publication test
      Number of rows in next upload: 0
      Last download: 1900-01-01 00:00:00.000
Done.
```

为名为 *sample.udb* 的文件将 `ml_remote_id` 设置为 NULL:

```
ulinfo -c DBF=sample.udb -rc
```

## UltraLite 初始化数据库实用程序 (ulinit)

从现有 SQL Anywhere 数据库创建 UltraLite 数据库。

### 语法

```
ulinit -a "SAconnection-string" -c "ULconnection-string" -n pubname [ options ]
```

选项	说明
<b>-a</b> "SAconnection-string"	必需。连接到 SAconnection-string 中指定的 SQL Anywhere 参考数据库。
<b>-c</b> "ULconnection-string"	必需。连接到 DBF 或 connection-string 的 file_name 参数所标识的数据库。如果未指定用户 ID 和口令，则会假定缺省 UID 为 DBA 且 PWD 为 sql。
<b>-d</b>	对于新 UltraLite 数据库中的各个表，将从 SQL Anywhere 数据库中相应表复制数据。缺省情况下，不将此数据上载到后续同步中。要在下一次上载同步中包含此数据，使用 <b>-I</b> 和 <b>-d</b> 。
<b>-e</b> table, ...	排除已命名的 table。已命名的表不是在 UltraLite 数据库中创建的。可在以逗号分隔的列表中指定多个表。例如：  <code>-e mydbtable1,mydbtable5</code>
<b>-I</b>	与 <b>-d</b> 配合使用。在下一次上载同步中包含已插入的行。缺省情况下，此实用程序插入的行不会在同步过程中上载。
<b>-l</b> logfile	执行时，将 DDL 数据库模式创建 SQL 语句记录到 logfile。
<b>-n</b> pubname	必需。将表添加到 UltraLite 数据库模式。  pubname 指定参考数据库中的发布。将发布中的表添加到 UltraLite 数据库中。  多次指定该选项可以将表从多个发布添加到 UltraLite 数据库。要将参考数据库中的所有表添加到 UltraLite 数据库中，指定 <b>-n*</b> 。
<b>-o</b> [ extended-options ]	指定 UltraLite 数据库创建参数的以分号分隔的列表。请参见“为 UltraLite 选择数据库创建参数”一节第 30 页。
<b>-p</b> creator-ID	将数据库安装到记录存储库时 Palm OS 所需要的选项。使用 UltraLite 客户端应用程序的指定四字符 creator-ID 创建数据库。如果您正在将数据库部署到 VFS 存储库，则无需使用此选项。
<b>-q</b>	将实用程序设置为在安静模式下运行。取消信息性标题、版本号和状态消息。但是，仍然会显示错误消息。

选项	说明
<b>-s</b> <i>pubname</i>	在 UltraLite 数据库中创建一个发布，与参考数据库一样定义为 <i>pubname</i> 。发布用于配置同步。提供多个 <b>-s</b> 选项以指定多个同步发布。 请注意，此发布中的表必须包含在用 <b>-n</b> 选项列出的发布中。 如果未提供 <b>-s</b> ，则 UltraLite 远程不会包含指定的发布。 有关如何为 MobiLink 同步创建发布的详细信息，请参见“ <a href="#">UltraLite 中的发布</a> ”一节第 124 页。
<b>-t</b> <i>file</i>	指定包含受信任的根证书的文件。服务器验证需要此证书。
<b>-w</b>	不显示警告。

## 注释

SQL Anywhere 参考数据库为以下各项充当源：

- 数据库配置（例如，所使用的归类序列）
- 表定义
- 同步发布

它们一起帮助创建 UltraLite 模式一定义新 UltraLite 数据库结构的信息。但是，创建的数据库初始为空。

如果不想使用 SQL Anywhere 参考数据库创建 UltraLite 数据库，请尝试以下方法之一：

- 如果想从 RDBMS 而不是 SQL Anywhere 初始化 UltraLite 数据库，则使用 Sybase Central 中的 [\[创建同步模型向导\]](#)。运行向导时，将提示您连接统一数据库以便获得模式信息。
- 如果想创建可以独立于任何种类的参考数据库进行配置的空 UltraLite 数据库，请使用 `ulcreate` 实用程序或 UltraLite 的 [\[创建数据库向导\]](#)。

UltraLite 使用在参考数据库中定义的归类序列的名称。但是，通过将 `utf8_encoding` 属性设置为 `extended-options` 列表中的一部分，仍可以选择使用 UTF-8 编码数据库。

要查看支持的归类（和相应代码页）的列表，在命令提示符下运行 `ulcreate -l`。如果 UltraLite 不支持您的归类序列，应该将其更改为 UltraLite 支持的归类序列。例如，如果您的参考数据库归类是 UCA 归类，则应该执行以下操作：

1. 卸载参考数据库，然后使用一种不同的归类重新进行装载。
2. 在这个新版数据库上运行 `ulinit`。

写入桌面操作系统的 Palm 数据库必须以 `.pdb` 扩展名标识。但是，一旦将数据库部署到设备，扩展名将被删除。有关文件名格式的详细信息，请参见“[Palm OS](#)”一节第 41 页。

如果目标是 VFS 卷，则不能使用 Palm 安装工具部署 UltraLite 数据库。而是，必须使用读卡器或某些其它工具将数据库直接复制到介质上。

## 另请参见

- “从 SQL Anywhere 参考数据库创建 UltraLite 数据库” 一节第 27 页
- “MobiLink 模型简介” 一节 《MobiLink - 入门》
- “UltraLite 创建数据库实用程序 (ulcreate)” 一节第 235 页
- “UltraLite 连接参数” 第 207 页

## 示例

创建名为 *customer.udb* 的文件，该文件包含 TestPublication 中定义的表：

```
ulinit -a "DSN=dbdsn;UID=JimmyB;PWD=secret" -c DBF=customer.udb -n  
TestPublication
```

创建名为 *customer.udb* 的文件，该文件包含两个不同的发布。具体地说，Pub1 可能包含用于优先级同步的一个小数据子集，而 Pub2 可能包含批量数据：

```
ulinit -a "DSN=dbdsn;UID=JimmyB;PWD=secret" -c DBF=customer.udb -n Pub1 -n  
Pub2 -s Pub1 -s Pub2
```

使用已注册的创建者 ID 为 Palm OS 创建名为 *customer.udb* 的文件：

```
ulinit -a "DSN=dbdsn;UID=JimmyB;PWD=secret" -c DBF=customer.udb.pdb -n  
TutCustomersPub -p creator-id
```

## UltraLite 装载 XML 到数据库实用程序 (ulload)

将 XML 文件的数据装载到一个新的或现有的数据库中。

### 语法

```
ulload -c "connection-string" [ options ] xml-file
```

选项	说明
<b>-a</b>	将数据和模式定义添加到现有数据库中。  如果要添加数据到预先存在的、基于记录的 Palm OS 数据库（带 .pdb 扩展名的数据库）中，则不要使用 -p 选项。
<b>-c "connection-string"</b>	必需。连接到 DBF 或 connection-string 的 file_name 参数所标识的数据库。如果未指定用户 ID 和口令，则会假定缺省 UID 为 <b>DBA</b> 且 PWD 为 <b>sql</b> 。
<b>-d</b>	只装载数据，忽略 XML 文件输入中的任何模式元数据。
<b>-f directory</b>	设置包含文件的目录，该文件带有要装载的附加数据。请参见 ulunload 中的 -f 选项（“UltraLite 卸载旧数据库实用程序 (ulunload)”一节第 260 页）。
<b>-g ID</b>	将初始数据库 ID 设置为所指派的 INTEGER 值。数据库将此值用于所有具有全局自动增量列的新行。数据库将此基值用于与每个附加行和/或列相关联的自动增量 ID。  在部署应用程序时，必须为每个数据库指派不同的标识号，目的是为了与 MobiLink 服务器同步。
<b>-I</b>	在下次上载同步中包含已插入的行。缺省情况下，此实用程序插入的行不会在同步过程中上载。
<b>-n</b>	只装载模式元数据，忽略 XML 输入文件中的任何数据。
<b>-o [ extended-options ]</b>	指定 UltraLite 数据库创建参数的以分号分隔的列表。请参见“为 UltraLite 选择数据库创建参数”一节第 30 页。
<b>-oa</b>	如果过程标识数据库由 UltraLite 的前一版本创建（将导致数据库升级），则将该过程取消。
<b>-ol</b>	列出可用的数据库创建参数，然后退出。请参见“为 UltraLite 选择数据库创建参数”一节第 30 页。

选项	说明
<b>-onerror behavior</b>	<p>控制在从 XML 文件中读取数据期间遇到错误时所采取的操作。指定以下受支持的 <i>behavior</i> 值之一：</p> <ul style="list-style-type: none"> <li>● <b>continue</b> uload 忽略错误并继续装载 XML。</li> <li>● <b>prompt</b> uload 会提示您输入信息，以了解是否要继续。</li> <li>● <b>quit</b> uload 停止装载 XML 并以一个错误终止运行。如果没有指定行为，则此项是缺省行为。</li> <li>● <b>exit</b> uload 退出。</li> </ul>
<b>-or</b>	<p>以只读模式打开数据库。UltraLite 会创建原始文件的副本，然后使用此副本测试脚本而不会变更数据库。完成后会放弃对复制文件的更改。</p> <p>如果直接从桌面操作系统连接到已部署到 Windows Mobile 设备的数据库，则不支持此选项。</p>
<b>-ou</b>	如果数据库由较旧版本的 UltraLite 创建，则升级数据库。
<b>-p creator-ID</b>	<p>将数据库安装到记录存储库时 Palm OS 所需要的选项。使用 UltraLite 客户端应用程序的指定四字符 <i>creator-ID</i> 创建数据库。如果您正在将数据库部署到 VFS 存储库，则无需使用此选项。</p> <p>如果要将数据添加到预先存在的、基于记录的 Palm OS 数据库（带有 <i>.pdb</i> 扩展名的数据库），则不要将此选项与 <b>-a</b> 选项一起使用。</p>
<b>-q</b>	将实用程序设置为在安静模式下运行。取消信息性标题、版本号和状态消息。但是，仍然会显示错误消息。
<b>-s file</b>	记录用于将数据库装载到指定的 <i>file</i> 的 SQL 语句。
<b>-t file</b>	指定包含受信任的根证书的文件。服务器验证需要此证书。
<b>-v</b>	打印详细消息。
<b>-y</b>	覆盖数据库文件，无须确认。这只适用于使用 uload 创建新数据库的情况。
<i>xml-file</i>	指定从其装载数据的 XML 文件的名称。

### 注释

uload 实用程序使输入 XML 文件由 ulunload、ulunloadold 或 ulxml 生成（在 UltraLite 版本 8 和本版本 9 中）。当与 ulunload 一起使用时，此实用程序可以提供重建数据库的功能。另一种重建数据库的方法是使用 ulunload 生成 SQL 语句，然后使用 DBISQL 将这些语句读入新数据库中。

XML 文件可以包含模式的元数据和/或数据库数据的元数据。**-d** 忽略模式元数据，只将数据添加到 *.udb* 文件。**-n** 忽略数据和元数据，只将模式添加到 *.udb* 文件。

在命令行中设置选项或指定证书将替换 uload 处理的 *xml-file* 的所有设置。

读取 XML 时，`ulload` 实用程序恢复数据库的所有同步配置文件。

此实用程序会返回错误代码。除 0 之外的任何值均意味着操作失败。

写入桌面操作系统的 Palm 数据库必须以 `.pdb` 扩展名标识。但是，一旦将数据库部署到设备，扩展名将被删除。有关文件名格式的详细信息，请参见“[Palm OS](#)”一节第 41 页。

如果目标是 VFS 卷，则不能使用 Palm 安装工具部署 UltraLite 数据库。而是，需要使用读卡器或某些其它工具将数据库直接复制到介质上。

### 另请参见

- [“注册 Palm 创建者 ID”一节 《UltraLite - C 及 C++ 编程》](#)
- [“UltraLite 连接参数”第 207 页](#)
- [“UltraLite 卸载旧数据库实用程序 \(ulunload\)”一节第 260 页](#)
- [“支持的退出代码”一节第 228 页](#)
- [“UltraLite global\\_database\\_id 选项”一节第 204 页](#)

### 示例

创建新的 UltraLite 数据库文件 `sample.udb`，并用 `sample.xml` 中的数据装载它：

```
ulload -c DBF=sample.udb sample.xml
```

将数据从 `sample.xml` 装载到现有数据库 `sample.udb`，如果发生错误，提示进行操作：

```
ulload -d -c DBF=sample.udb -onerror prompt sample.xml
```

将 XML 从名为 `test_data.xml` 的文件装载到 UltraLite 创建的、名为 `sample.udb` 的数据库的副本中。完成后放弃这些更改。此文件允许您检查 XML 数据中的错误，并更正它们。如果数据成功装载，可以运行不带 `-or` 选项的命令，以保持 XML 的更新：

```
ulload -or -c DBF=sample.udb -a test_data.xml
```



## UltraLite 同步实用程序 (ulsync)

将 UltraLite 数据库与 MobiLink 服务器同步。此工具可用于在应用程序开发中测试同步。

### 语法

**ulsync -c** "connection-string" [ options ] [ synchronization parameters { REPLACE | MERGE } profile-string ]

选项	说明
<b>-c</b> "connection-string"	必需。连接到 DBF 或 <i>connection-string</i> 的 <i>file_name</i> 参数所标识的数据库。如果未指定用户 ID 和口令，则会假定缺省 UID 为 <b>DBA</b> 且 PWD 为 <b>sql</b> 。
<b>-oa</b>	如果过程标识数据库由 UltraLite 的前一版本创建（将导致数据库升级），则将该过程取消。
<b>-or</b>	以只读模式同步数据库。UltraLite 会创建原始文件的副本，然后使用此副本测试脚本而不会变更数据库。完成后会放弃对复制文件的更改。  如果直接从桌面操作系统连接到已部署到 Windows Mobile 设备的数据库，则不支持此参数。
<b>-ou</b>	如果数据库由较旧版本的 UltraLite 创建，则升级数据库。
<b>-p</b>	使用已命名的同步配置文件同步等效于：  <pre>synchronize profileName merge 'syncOptions'</pre> 其中，同步选项从尾随 <b>ulsync</b> 选项中获取。例如：  <pre>ulsync -p profileName "MobiLinkUid=ml;ScriptVersion=Version001...syncOptions"</pre> 请参见“同步配置文件选项”一节第 258 页。
<b>-q</b>	将实用程序设置为在安静模式下运行。取消信息性标题、版本号和状态消息。但是，仍然会显示错误消息。
<b>-r</b>	显示上次同步的结果，然后退出。
<b>-s</b>	同步之后运行可用的 SQL 直通脚本。
<b>-v</b>	显示同步进度消息。还可以确定是否为任何同步显示进度，是否使用 C++ API 或 SQL 同步配置文件语句。请参见“UltraLite CREATE SYNCHRONIZATION PROFILE 语句”一节第 436 页。

## 注释

不再支持以下选项（它们对于版本 10 及更旧版本有效）：**-a authenticate-parameters**、**-e sync-parms**、**-k stream-type**、**-n**（不同步）和 **-x protocol options**。**-e <keyword>=<value>** 目前是同步参数字符串的一部分，并且 **-k** 和 **-x** 已包含于 **Stream=<stream{<stream-parms>}** 同步参数字符串。

根据用法，`ulsync` 被视为等效于以下 SQL 语句之一：

```
ulsync -p <profile> "<parms>"
```

等效于：

```
SYNCHRONIZE PROFILE <profile> MERGE
<parms>
```

和

```
ulsync "<parms>"
```

等效于：

```
SYNCHRONIZE USING <parms>
```

为了安全同步，UltraLite 应用程序必须已经访问了公共证书。可通过以下方法参考证书：

- 在创建时，使用 `ulinit`、`ulload` 或 `ulcreate`，通过 **-t file** 选项将证书信息加入到 UltraLite 数据库中。
- 在同步时通过 **trusted\_certificates=file** 流选项参考外部证书文件。

此实用程序会返回错误代码。除 0 之外的任何值均意味着操作失败。

## 另请参见

- “同步配置文件选项”一节第 258 页
- “UltraLite 连接参数”第 207 页
- “UltraLite 客户端”第 115 页
- “支持的退出代码”一节第 228 页
- “MobiLink 文件传输实用程序 (mlfiletransfer)”一节 《MobiLink - 客户端管理》
- “通过启用 TLS 的同步部署 UltraLite”一节第 51 页

## 示例

以下命令为名为一个 **remoteA** 的 MobiLink 用户同步一份名为 `myuldb.udb` 的数据库文件。

```
ulsync -c DBF=myuldb.udb "MobiLinkUid=remoteA;Stream=http;ScriptVersion=2"
```

以下命令通过带 `c:\certs\rsa.crt` 证书的 HTTPS 同步一份名为 `myuldb.udb` 的数据库文件。必须使用 **trusted\_certificates=file** 选项，因为在创建数据库时，未将受信任证书文件添加到数据库。此外，MobiLink 用户名是 **remoteB**。

```
ulsync -c DBF=myuldb.udb "Stream=https{trusted_certificates=c:\certs\rsa.crt};
MobiLinkUid=remoteB;ScriptVersion=2;UploadOnly=ON"
```

以下命令显示名为 `synced.udb` 的数据库文件的上一次同步的结果。

```
ulsync -r -c dbf=synced.udb
```

下面列出以前的同步结果:

```
SQL Anywhere UltraLite Database Synchronize Utility Version XX.X
Results of last synchronization:
Succeeded
  Download timestamp: 2006-07-25 16:39:36.708000
  Upload OK
  No ignored rows
  Partial download retained
  Authentication value: 1000 (0x3e8)
```

以下示例显示了在 2439 端口上, 通过 TCP/IP, 使用用户名 50 同步 CustDB 数据库的命令行。

```
ulsync -c "dbf=C:\Documents and Settings\All Users\Documents\SQL Anywhere
11\Samples\UltraLite\SyncEncrypt\custdb.udb"
MobiLinkId=50;ScriptVersion=custdb 11.0;Stream=tcpip{port=2439}
```

## 同步配置文件选项

在已定义要使用的所有其它命令行选项之后，在命令行上使用 `ulsync` 实用程序指定同步配置文件选项。关键字不区分大小写。

同步配置文件选项	有效值	说明
AllowDownloadDupRows	布尔值	下载具有相同主键的多行时，本选项可防止错误的出现。此选项可用于同步不一致的数据，而不会导致同步失败。缺省值为 "no"。请参见“ <a href="#">Additional Parameters 同步参数</a> ”一节第 138 页。
AuthParms	字符串 (以逗号分隔)	指定验证参数列表发送到 MobiLink 服务器。可使用验证参数在 MobiLink 脚本中执行自定义验证。请参见“ <a href="#">Authentication Parameters 同步参数</a> ”一节第 140 页。
CheckpointStore	布尔值	添加数据库在同步过程中的额外检查点，以限制数据库在同步过程中的增长。请参见“ <a href="#">Additional Parameters 同步参数</a> ”一节第 138 页。
ContinueDownload	布尔值	重新启动以前执行失败的下载。继续下载时，仅能接收到在同步失败时所选择要下载的更改。缺省情况下，UltraLite 不继续下载。请参见“ <a href="#">恢复失败的下载</a> ”一节《 <a href="#">MobiLink - 服务器管理</a> 》。
DisableConcurrency	布尔值	在同步期间禁止从其它线程访问数据库。请参见“ <a href="#">Additional Parameters 同步参数</a> ”一节第 138 页。
DownloadOnly	布尔值	执行仅下载同步。请参见“ <a href="#">Download Only 同步参数</a> ”一节第 142 页。
KeepPartialDownload	布尔值	如果发生通信错误，此选项控制 UltraLite 是否保持部分下载。缺省情况下，UltraLite 不回退部分下载的更改。请参见“ <a href="#">Keep Partial Download 同步参数</a> ”一节第 144 页。
MobiLinkPwd	字符串	指定与用户名关联的现有 MobiLink 口令。请参见“ <a href="#">MobiLinkPwd (mp) 扩展选项</a> ”一节《 <a href="#">MobiLink - 客户端管理</a> 》。
MobiLinkUid	字符串	指定 MobiLink 用户名。请参见“ <a href="#">-u 选项</a> ”一节《 <a href="#">MobiLink - 客户端管理</a> 》。请参见“ <a href="#">-mn 选项</a> ”一节《 <a href="#">MobiLink - 客户端管理</a> 》。
NewMobiLinkPwd	字符串	为 MobiLink 用户提供新口令。在想要更改现有口令时，可使用此选项。请参见“ <a href="#">-mn 选项</a> ”一节《 <a href="#">MobiLink - 客户端管理</a> 》。

同步配置文件选项	有效值	说明
Ping	布尔值	仅确认与服务器的通信，未执行同步。请参见“ <a href="#">Ping 同步参数</a> ”一节第 148 页。
发布	字符串 (以逗号分隔)	指定要同步的发布。发布可确定远程数据库上包含在同步操作中的表。如果此参数为空白（缺省值），则所有表都是同步的。如果此参数为星号(*)，则所有发布都是同步的。请参见“ <a href="#">UltraLite 中的发布</a> ”一节第 124 页。
ScriptVersion	字符串	指定 MobiLink 脚本版本。脚本版本确定在同步期间 MobiLink 在统一数据库上运行的脚本。如果未指定脚本版本，则使用“缺省值”。请参见“ <a href="#">ScriptVersion (sv) 扩展选项</a> ”一节《 <a href="#">MobiLink - 客户端管理</a> 》。
SendColumnNames	字符串	同步时应指定将列名作为上载文件的一部分发送到 MobiLink 服务器。缺省情况下，不发送列名。请参见“ <a href="#">Send Column Names 同步参数</a> ”一节第 151 页。
SendDownloadACK	布尔值	指定客户端应该向服务器发送一个下载确认。缺省情况下，MobiLink 服务器不提供下载确认。请参见“ <a href="#">发送下载确认同步参数</a> ”一节第 152 页。
stream	字符串 (带有 sub-list)	指定 MobiLink 网络同步协议。请参见“ <a href="#">Stream Type 同步参数</a> ”一节第 154 页。
TableOrder	字符串 (以逗号分隔)	指定上载中的表顺序。缺省情况下，UltraLite 选择基于外键关系的顺序。请参见“ <a href="#">Additional Parameters 同步参数</a> ”一节第 138 页。
UploadOnly	字符串	指定同步仅包含上载，不进行任何下载。请参见“ <a href="#">Upload Only 同步参数</a> ”一节第 157 页。

布尔值可被指定为 Yes/No、1/0、True/False、On/Off。在所有的布尔实例中，缺省值为 "No"。对于所有其它值，完全未指定缺省值。

#### 另请参见

- “[UltraLite ALTER SYNCHRONIZATION PROFILE 语句](#)”一节第 426 页
- “[UltraLite DROP SYNCHRONIZATION PROFILE 语句](#)”一节第 447 页
- “[UltraLite SYNCHRONIZE 语句](#)”一节第 462 页
- “[UltraLite 创建参数](#)”第 165 页

## UltraLite 卸载旧数据库实用程序 (ulunload)

根据所使用的选项，卸载以下内容：

- 将整个 UltraLite 数据库卸载到 XML 或 SQL。
- 只将全部或部分 UltraLite 数据卸载到 XML 或 SQL。

### 语法

**ulunload -c "connection-string" [ options ] output-file**

选项	说明
<b>-b max-size</b>	设置要存储到 XML 文件中的列数据的最大大小。缺省值为 10 KB。要将所有数据存储在 XML 文件（无最大大小）中，使用 <b>-b -1</b> 。
<b>-c "connection-string"</b>	必需。连接到 DBF 或 <i>connection-string</i> 的 <i>file_name</i> 参数所标识的数据库。如果未指定用户 ID 和口令，则会假定缺省 UID 为 <b>DBA</b> 且 PWD 为 <b>sql</b> 。
<b>-d</b>	只将数据库中的数据卸载到输出文件中。不卸载任何模式信息。
<b>-e table, ...</b>	卸载数据库时排除已命名 <i>table</i> 。可在以逗号分隔的列表中指定多个表。例如： <code>-e mydbtable1,mydbtable5</code>
<b>-f directory</b>	将存储数据的目录设置为大于由 <b>-b</b> 指定的最大大小。缺省情况下与输出文件的目录相同。
<b>-n</b>	只卸载模式，忽略数据库中的所有数据。
<b>-oa</b>	如果过程标识数据库由 UltraLite 的前一版本创建（将导致数据库升级），则将该过程取消。
<b>-or</b>	以只读模式打开数据库。UltraLite 备份了原始文件并使用副本进行卸载。这可以确保使用软件早期版本创建的数据库不会被升级。 如果直接从桌面操作系统连接到已部署到 Windows Mobile 设备的数据库，则不支持此参数。
<b>-ou</b>	如果数据库由较旧版本的 UltraLite 创建，则升级数据库。
<b>-q</b>	将实用程序设置为在安静模式下运行。取消信息性标题、版本号和状态消息。但是，仍然会显示错误消息。
<b>-s</b>	作为与 SQL Anywhere 兼容的 SQL 语句卸载。SQL 文件输出可使用 DBISQL，由 UltraLite 或 SQL Anywhere 来读取。
<b>-t table, ...</b>	只卸载指定的 <i>table</i> 中的数据。可在以逗号分隔的列表中指定多个表。例如： <code>-t mydbtable2,mydbtable6</code>

选项	说明
<code>-v</code>	打印详细消息。
<code>-x owner</code>	输出表，以使这些表由特定用户 ID 拥有。可以将此选项与 <code>-s</code> 选项一起使用。
<code>-y</code>	覆盖 <i>output-file</i> 而无需确认。
<i>output-file</i>	必需。设置数据库将卸载到的文件的名称。如果使用 <code>-s</code> 选项，数据库将卸载为 SQL 语句。否则，数据库将卸载为 XML。

## 注释

缺省情况下，ulunload 将输出描述数据库中的模式和数据的 XML。可将此输出用于归档目的，或用于保持 UltraLite 数据库在所有版本之间的可移植性。

将带同步配置文件结果的数据库保存到与 UltraLite 实用程序的早期版本不兼容的 XML 中。解决方案是编辑 XML 并移除文本区段，该文本区段标记为

```
<syncprofiles>...</syncprofiles>
```

卸载数据库不会保留：

- 同步状态、存储的同步计数和行删除。请确保在卸载之前同步数据库。
- UltraLite 用户条目。

要确认保留了哪些数据库选项或属性，请在使用 ulload 实用程序重装数据库以后运行 ulinfo。

如果列数据超过用 `-b` 指定的最大大小，溢出将保存在一个 *\*.bin* 文件中，该文件位于：

- 与 XML 文件相同的目录中
- 由 `-f` 指定的目录中。

该文件遵循以下命名约定：

```
tablename-columnname-rownumber.bin
```

`-x` 选项允许您为 UltraLite 表指派所有权。如果打算使用结果 SQL 语句创建或修改 SQL Anywhere 数据库，只需要为表指派一个所有者。由 UltraLite 读取时，所有者名称以静默方式被忽略。

此实用程序会返回错误代码。除 0 之外的任何值均意味着操作失败。

如果正在使用此实用程序直接卸载 Windows Mobile 设备上的数据库，则 UltraLite 在卸载或操作发生之前无法备份数据库。必须先手工执行此操作，然后才能运行这些向导。

## 另请参见

- “UltraLite 连接参数” 第 207 页
- “支持的退出代码” 一节第 228 页
- “UltraLite 装载 XML 到数据库实用程序 (ulload)” 一节第 252 页
- “UltraLite 信息实用程序 (ulinfo)” 一节第 245 页

## 示例

将 *sample.udb* 数据库卸载到 *sample.xml* 文件中。

```
ulunload -c DBF=sample.udb sample.xml
```

将 *sample.udb* 数据库中的数据卸载到名为 *sample1.sql* 的 SQL 文件中。如果 SQL 文件存在，则将其覆盖。

```
ulunload -c DBF=sample.udb -d -y sample.sql
```



## UltraLite 卸载旧数据库实用程序 (ulunloadold)

将 UltraLite 版本 8.0.2 至版本 9.0.x 的数据库和/或模式文件 (\*.usm) 卸载到 XML 文件。

### 语法

```
ulunloadold -c "connection-string" [ options ] xml-file
```

选项	说明
<b>-b</b> <i>max-size</i>	设置要存储到 XML 文件中的列数据的最大大小。缺省值为 10 KB。要将所有数据存储在 XML 文件（无最大大小）中，使用 <b>-b -1</b> 。
<b>-c</b> " <i>connection-string</i> "	必需。连接到 DBF 或 <i>connection-string</i> 的 <i>file_name</i> 参数所标识的数据库。如果未指定用户 ID 和口令，则会假定缺省 UID 为 <b>DBA</b> 且 PWD 为 <b>sql</b> 。
<b>-f</b> <i>directory</i>	将存储数据的目录设置为大于由 <b>-b</b> 指定的最大大小。缺省情况下与 XML 文件的目录相同。
<b>-q</b>	将实用程序设置为在安静模式下运行。取消信息性标题、版本号和状态消息。但是，仍然会显示错误消息。
<b>-v</b>	打印详细消息。
<b>-y</b>	覆盖 <i>xml-file</i> 而无需确认。
<i>xml-file</i>	设置数据将卸载到的 XML 文件的名称。

### 注释

UltraLite 版本 11 不能直接升级版本 8.x 或版本 9.x 数据库。使用此工具生成 XML 文件，然后，ulload 使用此文件创建版本 11 数据库。请不要用此实用程序卸载 UltraLite 版本 11 数据库。请使用 ulunload 实用程序。

卸载数据库不会保留：

- 同步状态、存储的同步计数和行删除。请确保在卸载之前同步数据库。
- UltraLite 用户条目。

要确认保留了哪些数据库选项或属性，请在使用 ulload 实用程序重装数据库以后运行 ulinfo。

如果列数据超过用 **-b** 指定的最大大小，溢出将保存在一个 \*.bin 文件中，该文件位于：

- 与 XML 文件相同的目录中
- 由 **-f** 指定的目录中

该文件名称遵循以下命名约定：

```
tablename-columnname-rownumber.bin
```

除 0 之外的任何错误代码值均意味着操作失败。

此实用程序不能用于在 Windows Mobile 设备上直接卸载数据库。必须先将这些数据库复制到台式计算机。

### 另请参见

- [“UltraLite 连接参数” 第 207 页](#)
- [“UltraLite 装载 XML 到数据库实用程序 \(ulload\)” 一节第 252 页](#)
- [“UltraLite 卸载旧数据库实用程序 \(ulunload\)” 一节第 260 页](#)
- [“UltraLite 信息实用程序 \(ulinfo\)” 一节第 245 页](#)

### 示例

将名为 *dbschema8.usm* 的 UltraLite 8.0.x 模式文件升级成为名为 *db.udb* 的 UltraLite 版本 11 数据库需要以下两条命令：

```
ulunloadold -c SCHEMA_FILE=dbschema8.usm dbschema.xml  
ulload -c DBF=db.udb dbschema.xml
```

将名为 *palm9db.pdb* 的 Palm OS UltraLite 版本 9.0.x 数据库升级成为名为 *palm11db.pdb* 的 UltraLite 版本 11 数据库需要以下两条命令：

```
ulunloadold -c DBF=palm9db.pdb dbdata.xml  
ulload -c DBF=palm11db.pdb -p Syb1 dbdata.xml
```

## UltraLite 校验数据库实用程序 (ulvalid)

执行 UltraLite 数据库的完全校验 ("normal"), 包括:

- 数据库页 - 启用时使用校验和校验所有数据库页 (请参见 `checksum_level` 数据库创建参数)。请注意, 某些关键页始终拥有校验和, 而不带校验和的页进行基本校验检查。
- 表 - 通过检查表行计数是否与各个索引中的计数相匹配来校验表。
- 索引 - 通过检查条目是否引用了有效行来校验索引。ulvalid -e 执行快速检查, 其中只包括表校验。

### 语法

`ulvalid -c "connection-string" [ options ]`

选项	说明
<code>-c "connection-string"</code>	必需。连接到在 <code>connection-string</code> 中标识的数据库。如果未指定用户 ID 和口令, 则会假定缺省 UID 为 <b>DBA</b> 且 PWD 为 <b>sql</b> 。
<code>-e</code>	快速校验。只执行表校验。此选项提供了一种比常规校验更快的校验。
<code>-oa</code>	如果过程标识数据库由 UltraLite 的前一版本创建 (将导致数据库升级), 则将该过程取消。
<code>-or</code>	以只读模式打开数据库。UltraLite 会创建原始文件的副本, 然后可以使用此副本测试脚本而不会变更数据库。完成后会放弃对复制文件的更改。 如果直接从桌面操作系统连接到已部署到 Windows Mobile 设备的数据库, 则不支持此参数。
<code>-ou</code>	如果数据库由较旧版本的 UltraLite 创建, 则升级数据库。
<code>-q</code>	将实用程序设置为在安静模式下运行。取消信息性标题、版本号和状态消息。但是, 仍然会显示错误消息。
<code>-v</code>	打印详细消息。

### 注释

校验数据库验证表元数据的准确性并确保文件尚未被损坏。

### 另请参见

- “校验 UltraLite 数据库” 一节第 12 页

### 示例

名为 `sample.udb` 的数据库的快速校验的示例以安静模式运行。

```
ulvalid -c DBF=sample.udb -e -q
```

---

---

# UltraLite 系统表

## 目录

查看或隐藏 UltraLite 系统表 .....	268
systable 系统表 .....	269
syscolumn 系统表 .....	270
sysindex 系统表 .....	271
sysixcol 系统表 .....	273
syspublication 系统表 .....	274
sysarticle 系统表 .....	275
sysuldata 系统表 .....	276

---

每个 UltraLite 数据库的模式都是以几个系统表来描述的。由于 UltraLite 不支持表所有权，因此任何 UltraLite 用户都可以访问这些系统表。

系统表的内容只能由 UltraLite 本身进行更改。不能使用 UPDATE、DELETE 和 INSERT 命令来修改这些表的内容。此外，也不能使用 ALTER TABLE 和 DROP 语句来更改这些表的结构。

## 查看或隐藏 UltraLite 系统表

### ◆ 隐藏或显示系统对象 (Sybase Central)

1. 连接到数据库。
2. 浏览数据库的对象。
3. 右击 [内容] 窗格并选择 [隐藏/显示系统对象]。
4. 单击 [表] 并浏览可用表。

## systable 系统表

systable 系统表中的每一行都描述了数据库中的一个表。

列名	列类型	说明
column_count	UNSIGNED INT	表中的列数。
index_count	UNSIGNED INT	表中的索引数。
indexcol_count	UNSIGNED INT	表中所有索引的总列数。
map_handle	UNSIGNED INT	仅供内部使用。
table_name	VARCHAR(128)	表的名称。
object_id	UNSIGNED INT	该表的唯一标识符。
sync_type	VARCHAR(128)	用于 MobiLink 同步。可以为以下任意一种：表示不同步的 <b>no_sync</b> 、表示同步每一行的 <b>all_sync</b> 或表示仅同步更改行的 <b>normal_sync</b> 。
table_name	VARCHAR(128)	表的名称。
table_type	TINYINT	既可以是表示系统表的 <b>sys</b> ，也可以是表示常规表的 <b>user</b> 。
tpd_handle	UNSIGNED INT	仅限内部使用者。

### 约束

PRIMARY KEY (object\_id)

## syscolumn 系统表

syscolumn 系统表中的每一行都描述了一列。

列名	列类型	说明
column_name	VARCHAR(128)	列的唯一标识符。
default	VARCHAR(128)	此列的缺省值。例如，自动增量。
domain	UNSIGNED INT	列域，用来指示列的域的枚举值。
domain_info	SMALLINT	用于大小可变的域。
nulls	CHAR(1)	确定列是否允许缺省值为空值。
object_id	UNSIGNED INT	该列的唯一标识符。
table_id	UNSIGNED INT	列所属表的标识符。

### 约束

PRIMARY KEY( table\_id, object\_id )

FOREIGN KEY (table\_id) REFERENCES systable (object\_id)



## sysindex 系统表

sysindex 系统表中的每一行都描述了数据库中的一个索引。

列名	列类型	说明
check_on_commit	UNSIGNED INT	指示何时检查参照完整性以确保每一个外键都有一个相匹配的主行。只有类型为 <b>foreign</b> 时才需要。
index_name	UNSIGNED INT	索引的名称。
ixcol_count	UNSIGNED INT	索引中的列数。
nullable	BIT	只有类型为 <b>foreign</b> 时才需要。指出是否允许空值。
object_id	UNSIGNED INT	索引的唯一标识符。
primary_index_id	UNSIGNED INT	只有类型为 <b>foreign</b> 时才需要。列出了主索引的标识符。
primary_table_id	UNSIGNED INT	只有类型为 <b>foreign</b> 时才需要。列出了主表的标识符。
root_handle	UNSIGNED INT	仅供内部使用。
table_id	UNSIGNED INT	应用索引的表的唯一标识符。
type	SMALLINT (10)	索引的类型。可以是以下值之一： <ul style="list-style-type: none"> <li>● <b>primary</b></li> <li>● <b>foreign</b></li> <li>● <b>key</b></li> <li>● <b>unique</b></li> <li>● <b>index</b></li> </ul>
hash_size	SHORT	存储用于索引散列的散列大小。

### 约束

PRIMARY KEY( table\_id, object\_id )

FOREIGN KEY (table\_id) REFERENCES systable (object\_id)

**另请参见**

- [“sysixcol 系统表”一节第 273 页](#)

## sysixcol 系统表

sysixcol 系统表中的每一行都描述了在 sysindex 中列出的索引的一个列。

列名	列类型	说明
column_id	UNSIGNED INT	建立索引列的唯一标识符。
index_id	UNSIGNED INT	该索引列所属索引的唯一标识符。
order	CHAR(1)	指出该列在索引中是升序 (A) 还是降序 (D)。
sequence	SMALLINT	索引中列的顺序。
table_id	UNSIGNED INT	应用索引的表的唯一标识符。

### 约束

PRIMARY KEY( table\_id, index\_id, sequence )

FOREIGN KEY( table\_id, index\_id ) REFERENCES sysindex( table\_id, object\_id )

FOREIGN KEY( table\_id, column\_id ) REFERENCES syscolumn( table\_id, object\_id )

### 另请参见

- “sysindex 系统表” 一节第 271 页

## syspublication 系统表

syspublication 系统表中的每一行都描述了一个发布。

列名	列类型	说明
download_timestamp	UNSIGNED INT	上次下载的时间。
last_sync	UNSIGNED BIGINT	用于跟踪上载进度。
publication_id	UNSIGNED INT	发布的唯一标识符。
publication_name	CHAR(128)	发布的名称。

### 约束

PRIMARY KEY (publication\_id)

### 另请参见

- [“sysarticle 系统表”一节第 275 页](#)

## sysarticle 系统表

sysarticle 系统表中的每一行都描述了属于发布的一个表。

列名	列类型	说明
publication_id	UNSIGNED INT	该项目所属的发布的标识符。
table_id	UNSIGNED INT	属于发布的表的标识符。
where_expr	TINY INT	可过滤行的可选谓语句。

### 约束

PRIMARY KEY (publication\_id, table\_id)

FOREIGN KEY (publication\_id) REFERENCES syspublication (publication\_id)

FOREIGN KEY (table\_id) REFERENCES systable (object\_id)

### 另请参见

- [“syspublication 系统表” 一节第 274 页](#)

## sysuldata 系统表

sysuldata 系统表中的每一行都指定了选项和属性的值对。

列名	列类型	说明
long_setting	LONGBINARY	long 型值的 BLOB。
name	VARCHAR(128)	属性的名称。
setting	VARCHAR(128)	属性值。
type	VARCHAR(128)	可以是表示内部项的 <b>sys</b> 、表示选项的 <b>opt</b> 或表示属性的 <b>prop</b>

### 约束

PRIMARY KEY (name, type)

### 另请参见

- [“UltraLite 数据库属性” 第 195 页](#)

# UltraLite SQL 参考

本节提供 UltraLite SQL 的参考资料。UltraLite SQL 是 SQL Anywhere 数据库所支持的 SQL 的唯一子集。

---

UltraLite SQL 元素 .....	279
UltraLite SQL 函数 .....	323
UltraLite SQL 语句 .....	421





---

# UltraLite SQL 元素

## 目录

UltraLite 中的关键字 .....	280
UltraLite 中的标识符 .....	281
UltraLite 中的字符串 .....	282
UltraLite 中的注释 .....	283
UltraLite 中的数字 .....	284
UltraLite 中的 NULL 值 .....	285
UltraLite 中的特殊值 .....	286
UltraLite 中的日期和时间 .....	289
UltraLite 中的数据类型 .....	290
UltraLite 中的表达式 .....	303
UltraLite 中的运算符 .....	316
UltraLite 中的变量 .....	319
UltraLite 中的执行计划 .....	320

## UltraLite 中的关键字

每个 SQL 语句包含一个或多个关键字。SQL 的关键字不区分大小写，但在这些手册中，关键字都以大写字母表示。有些关键字只有用双引号引起来才能用作标识符。这些关键字被称为 **reserved words**。请参见“保留字”一节《SQL Anywhere 服务器 - SQL 参考》。

**注意**

UltraLite 仅支持 SQL Anywhere 关键字的子集。但是为了避免在将来的版本中出现潜在的问题，应假定 SQL Anywhere 的所有保留字也应用为 UltraLite。

## UltraLite 中的标识符

**Identifiers** 是数据库中对象（如用户 ID、表和列）的名称。标识符的最大长度为 128 个字节。

当以下任一条件成立时，必须将标识符用双引号引起来：

- 标识符包含空格。
- 标识符的首字符不是字母字符。数据库归类序列指出了哪些字符被视为字母字符或数字字符。
- 标识符包含保留字。请参见“保留字”一节《[SQL Anywhere 服务器 - SQL 参考](#)》。
- 标识符包含字母和数字以外的其它字符。

只有当作为转义字符使用时，才能在标识符中使用单反斜线。

## UltraLite 中的字符串

字符串用于保存数据库中的字符数据。UltraLite 支持的字符串规则与 SQL Anywhere 支持的字符串规则相同。字符串的比较结果和排序顺序取决于数据库是否区分大小写、字符集和归类序列。这些属性是在创建数据库时设置的。

### 另请参见

- “字符串”一节 《SQL Anywhere 服务器 - SQL 参考》
- “UltraLite 字符集”一节第 32 页

## UltraLite 中的注释

注释用于在 SQL 语句或语句块中附加说明性文字。UltraLite 运行时不执行注释。

UltraLite 中使用以下几种注释指示符：

- **-- (双连字符)** 数据库服务器忽略行上的任何其余字符。该指示符是 SQL/2003 注释指示符。
- **// (双斜线)** 双斜线与双连字符的含义相同。
- **/\* ... \*/ (斜线加星号)** 忽略两个注释标记间的任何字符。两个注释标记可以在同一行上或者在不同的行上。可以嵌套以此风格指示的注释。这种注释风格也称为 C 样式注释。

### 注意

UltraLite 不支持百分号 (%)。

### 示例

- 以下示例说明了如何使用双连字符注释：

```
CREATE TABLE borrowed_book (
  loaner_name CHAR(100)      PRIMARY KEY,
  date_borrowed DATE NOT NULL DEFAULT CURRENT DATE,
  date_returned      DATE,
  book               CHAR(20)
  FOREIGN KEY book REFERENCES library_books (isbn),
);
--This statement creates a table for a library database to hold information
on borrowed books.
--The default value for date_borrowed indicates that the book is borrowed
on the day the entry is made.
--The date_returned column is NULL until the book is returned.
```

- 以下示例说明如何使用 C 样式注释：

```
CREATE TABLE borrowed_book (
  loaner_name CHAR(100)      PRIMARY KEY,
  date_borrowed DATE NOT NULL DEFAULT CURRENT DATE,
  date_returned      DATE,
  book               CHAR(20)
  FOREIGN KEY book REFERENCES library_books (isbn),
);
/* This statement creates a table for a library database to hold
information on borrowed books.
The default value for date_borrowed indicates that the book is borrowed on
the day the entry is made.
The date_returned column is NULL until the book is returned. */
```

## UltraLite 中的数字

数字用于保存数据库中的数字数据。数字可以：

- 是任意数字序列
- 附加有小数部分
- 包含可选负号 (-) 或正号 (+)
- 跟有一个 e 和一个数字指数值

例如，UltraLite 支持以下所示的全部数字：

42

-4.038

.001

3.4e10

1e-10

## UltraLite 中的 NULL 值

与在 SQL Anywhere 中的情况一样，NULL 是一个特殊值，它不同于任何数据类型的任何有效值。然而，NULL 值在任何数据类型中都是合法值。NULL 用于表示未知（没有值）或不适用的信息。请参见“NULL 值”一节 [《SQL Anywhere 服务器 - SQL 参考》](#)。

## UltraLite 中的特殊值

特殊值可用在表达式中，也可在创建表时用作列缺省值。

### CURRENT DATE 特殊值

返回当前年月日。

#### 数据类型

DATE

#### 注释

当此 SQL 语句由 UltraLite 运行时执行时，返回的日期基于系统时钟的读数。如果在以下各种情况下使用 CURRENT DATE，全部值会基于独立的时钟读数：

- 同一语句中多次出现 CURRENT DATE
- 在一个语句中 CURRENT DATE 与 CURRENT TIME 或 CURRENT TIMESTAMP 一起使用
- 在一个语句中 CURRENT DATE 与 NOW 函数或 GETDATE 函数一起使用

#### 另请参见

- [“UltraLite 中的表达式”一节第 303 页](#)
- [“GETDATE 函数 \[Date and time\]”一节第 361 页](#)
- [“NOW 函数 \[Date and time\]”一节第 384 页](#)

### CURRENT TIME 特殊值

返回当前小时、分钟、秒和秒的小数部分。

#### 数据类型

TIME

#### 注释

秒的小数部分存储到 6 个小数位。当前时间的准确性受系统时钟准确性的限制。

当此 SQL 语句由 UltraLite 运行时执行时，返回的日期基于系统时钟的读数。如果在以下各种情况下使用 CURRENT TIME，全部值会基于独立的时钟读数：

- 同一语句中多次出现 CURRENT TIME
- 在一个语句中 CURRENT TIME 与 CURRENT DATE 或 CURRENT TIMESTAMP 一起使用
- 在一个语句中 CURRENT TIME 与 NOW 函数或 GETDATE 函数一起使用



**另请参见**

- “UltraLite 中的表达式” 一节第 303 页
- “GETDATE 函数 [Date and time]” 一节第 361 页
- “NOW 函数 [Date and time]” 一节第 384 页

## CURRENT\_TIMESTAMP 特殊值

组合 CURRENT DATE 和 CURRENT TIME，以构成包含年、月、日、小时、分钟、秒和秒的小数部分的 TIMESTAMP 值。

**数据类型**

TIMESTAMP

**注释**

秒的小数部分存储到 3 个小数位。其准确性受系统时钟准确性的限制。

以 DEFAULT CURRENT\_TIMESTAMP 声明的列不必包含唯一值。

CURRENT\_TIMESTAMP 返回的信息与 GETDATE 和 NOW 函数返回的信息相同。

CURRENT\_TIMESTAMP 与 CURRENT\_TIMESTAMP 等价。

当此 SQL 语句由 UltraLite 运行时执行时，返回的日期基于系统时钟的读数。如果在以下各种情况下使用 CURRENT\_TIMESTAMP，全部值会基于独立的时钟读数：

- 同一语句中多次出现 CURRENT\_TIMESTAMP
- 在一个语句中 CURRENT\_TIMESTAMP 与 CURRENT DATE 或 CURRENT TIME 一起使用
- 在一个语句中 CURRENT\_TIMESTAMP 与 NOW 函数或 GETDATE 函数一起使用

**另请参见**

- “CURRENT TIME 特殊值” 一节第 286 页
- “UltraLite 中的表达式” 一节第 303 页
- “NOW 函数 [Date and time]” 一节第 384 页
- “GETDATE 函数 [Date and time]” 一节第 361 页
- “NOW 函数 [Date and time]” 一节第 384 页

## SQLCODE 特殊值

计算特殊值时的当前 SQLCODE 值。

**数据类型**

String

### 注释

SQLCODE 值在每个语句后设置。可以检查 SQLCODE 来确定语句是否成功。

### 另请参见

- [“UltraLite 中的表达式” 一节第 303 页](#)
- [错误消息](#)

### 示例

使用 SELECT 语句为从结果集读取新行的每次尝试生成错误代码。例如：`SELECT a, b, SQLCODE FROM MyTable`。

## UltraLite 中的日期和时间

很多日期和时间函数使用从日期和时间部分生成的日期。UltraLite 和 SQL Anywhere 支持相同的日期部分。请参见“日期部分”一节第 325 页。

## UltraLite 中的数据类型

UltraLite SQL 中的可用数据类型包括:

- 整数
- 十进制
- 浮点
- 字符
- 二进制
- 日期/时间

### 注意

域（用户定义的数据类型）在 UltraLite SQL 中不受支持。

### 注意

不能连接 LONGVARCHAR 和 LONGBINARY 数据类型。请参见“字符串运算符”一节第 317 页。

可使用任何一种受支持的类型创建主机变量。UltraLite 支持 SQL Anywhere 中可用数据类型的子集。下面是 UltraLite 数据库中支持的 SQL 数据类型。

数据类型	说明
<b>BIT</b>	布尔值（0 或 1）。请参见“ <a href="#">BIT 数据类型</a> ”一节《 <a href="#">SQL Anywhere 服务器 - SQL 参考</a> 》。
{ <b>CHAR</b>   <b>CHARACTER</b> } ( <i>max-length</i> )	最大长度为 <i>max-length</i> 的字符数据，范围在 1-32767 字节之间。请参见“ <a href="#">CHAR 数据类型</a> ”一节《 <a href="#">SQL Anywhere 服务器 - SQL 参考</a> 》。 计算表达式时，临时字符值的最大长度为 2048 字节。
<b>VARCHAR</b> ( <i>max-length</i> )	<b>VARCHAR</b> 用于最大长度为 <i>max-length</i> 的可变长度字符数据。请参见“ <a href="#">VARCHAR 数据类型</a> ”一节《 <a href="#">SQL Anywhere 服务器 - SQL 参考</a> 》。
<b>LONG VARCHAR</b>	任意长度的字符数据。SQL 语句中的条件（例如，在 WHERE 子句中）不能对 <b>LONG VARCHAR</b> 列进行操作。允许对 <b>LONG VARCHAR</b> 列执行的操作只有插入、更新或删除，或将其包含在查询的 <i>select-list</i> 中。请参见“ <a href="#">LONG VARCHAR 数据类型</a> ”一节《 <a href="#">SQL Anywhere 服务器 - SQL 参考</a> 》。 您可以将字符串与 <b>LONGVARCHAR</b> 数据进行相互转换。
[ <b>UNSIGNED</b> ] <b>BIGINT</b>	需要 8 个字节存储空间的整数。请参见“ <a href="#">BIGINT 数据类型</a> ”一节《 <a href="#">SQL Anywhere 服务器 - SQL 参考</a> 》。

数据类型	说明
{ <b>DECIMAL</b>   <b>DEC</b>   <b>NUMERIC</b> } ( <i>precision</i> , <i>scale</i> ) ] ]	用两部分表示十进制数字: <i>precision</i> (总位数) 和 <i>scale</i> (小数点后的位数)。请参见“ <a href="#">DECIMAL 数据类型</a> ”一节《 <a href="#">SQL Anywhere 服务器 - SQL 参考</a> 》、“ <a href="#">NUMERIC 数据类型</a> ”一节《 <a href="#">SQL Anywhere 服务器 - SQL 参考</a> 》、“ <a href="#">UltraLite 精度创建参数</a> ”一节第 184 页和“ <a href="#">UltraLite 小数位数创建参数</a> ”一节第 185 页。
<b>DOUBLE</b> [ <b>PRECISION</b> ]	双精度浮点数。在此数据类型中, PRECISION 是 DOUBLE 数据类型名称的可选部分。请参见“ <a href="#">DOUBLE 数据类型</a> ”一节《 <a href="#">SQL Anywhere 服务器 - SQL 参考</a> 》。
<b>FLOAT</b> [ ( <i>precision</i> ) ]	浮点数, 单精度或双精度均可。请参见“ <a href="#">FLOAT 数据类型</a> ”一节《 <a href="#">SQL Anywhere 服务器 - SQL 参考</a> 》。
[ <b>UNSIGNED</b> ] { <b>INT</b>   <b>INTEGER</b> }	需要 4 个字节存储空间的无符号整数。请参见“ <a href="#">INTEGER 数据类型</a> ”一节《 <a href="#">SQL Anywhere 服务器 - SQL 参考</a> 》。
<b>REAL</b>	以 4 个字节存储的单精度浮点数。请参见“ <a href="#">REAL 数据类型</a> ”一节《 <a href="#">SQL Anywhere 服务器 - SQL 参考</a> 》。
[ <b>UNSIGNED</b> ] <b>SMALLINT</b>	需要 2 个字节存储空间的整数。请参见“ <a href="#">SMALLINT 数据类型</a> ”一节《 <a href="#">SQL Anywhere 服务器 - SQL 参考</a> 》。
[ <b>UNSIGNED</b> ] <b>TINYINT</b>	需要 1 个字节存储空间的整数。请参见“ <a href="#">TINYINT 数据类型</a> ”一节《 <a href="#">SQL Anywhere 服务器 - SQL 参考</a> 》。
<b>DATE</b>	日历日期, 如年、月和日。请参见“ <a href="#">DATE 数据类型</a> ”一节《 <a href="#">SQL Anywhere 服务器 - SQL 参考</a> 》。
<b>TIME</b>	某天的时间, 包括小时、分钟、秒和秒的小数部分。请参见“ <a href="#">TIME 数据类型</a> ”一节《 <a href="#">SQL Anywhere 服务器 - SQL 参考</a> 》。
<b>DATETIME</b>	与 <b>TIMESTAMP</b> 相同。请参见“ <a href="#">DATETIME 数据类型</a> ”一节《 <a href="#">SQL Anywhere 服务器 - SQL 参考</a> 》。
<b>TIMESTAMP</b>	时间点, 包括年、月、日、小时、分钟、秒和秒的小数部分。请参见“ <a href="#">TIMESTAMP 数据类型</a> ”一节《 <a href="#">SQL Anywhere 服务器 - SQL 参考</a> 》。
<b>VARBINARY</b> ( <i>max-length</i> )	与 <b>BINARY</b> 相同。请参见“ <a href="#">VARBINARY 数据类型</a> ”一节《 <a href="#">SQL Anywhere 服务器 - SQL 参考</a> 》。
<b>BINARY</b> ( <i>max-length</i> )	最大长度为 <i>max-length</i> 个字节的二进制数据。最大长度不应超过 2048 个字节。请参见“ <a href="#">BINARY 数据类型</a> ”一节《 <a href="#">SQL Anywhere 服务器 - SQL 参考</a> 》。

数据类型	说明
<b>LONG BINARY</b>	任意长度的二进制数据。SQL 语句中的条件（例如，在 WHERE 子句中）不能对 LONG BINARY 列进行操作。允许对 LONG BINARY 列执行的操作只有插入、更新或删除，或将其包含在查询的 <i>select-list</i> 中。请参见“ <a href="#">LONG BINARY 数据类型</a> ”一节《 <a href="#">SQL Anywhere 服务器 - SQL 参考</a> 》。 您可以将值与 LONGVARCHAR 数据相互转换。
<b>UNIQUEIDENTIFIER</b>	通常用于主键或其它唯一列以保存唯一标识行的 UUID（通用唯一标识符）值。UltraLite 提供生成 UUID 值的函数。在一台计算机上所生成的值与在其它计算机上生成的 UUID 值不相同。因此，以这种方式生成的 UNIQUEIDENTIFIER 值可用作同步环境中的键。请参见“ <a href="#">UNIQUEIDENTIFIER 数据类型</a> ”一节《 <a href="#">SQL Anywhere 服务器 - SQL 参考</a> 》。

## 用户定义的数据类型及其等效项

与 SQL Anywhere 数据库不同，UltraLite 不支持用户定义的数据类型。下表列出了等效于内置 SQL Anywhere 别名的 UltraLite 数据类型：

SQL Anywhere 数据类型	UltraLite 等效项
MONEY	NUMERIC(19,4)
SMALLMONEY	NUMERIC(10,4)
TEXT	LONG VARCHAR
XML	LONG VARCHAR

## 显式转换数据类型

UltraLite 允许使您使用 CAST 或 CONVERT 函数显式地请求数据类型转换。

### 注意

大多数情况下，自转换对操作没有影响。但是，自转换到 CHAR/VARCHAR、BINARY/VARBINARY 和 NUMERIC 不是空操作过程。

可以 CAST 或 CONVERT 大多数数据类型的组合，如下表所示。

但在某些情况下，能否转换取决于转换中使用的值。因为 **Value-dependent** 列显示的值必须与新数据类型兼容，以避免生成特定类型的转换错误。例如：

- 如果将 **varchar "1234"** 转换为 **long**，此转换会受到支持。但是，如果将 **varchar "hello"** 转换为 **long**，那么此转换会生成 **SQL\_CONVERT\_ERROR** 错误，因为 **hello** 并不是一个数字。
- 如果将 **long 1234** 转换为 **short**，此转换会受到支持。但是，如果将 **long 100000** 转换为 **short**，那么此转换会生成 **SQL\_OVERFLOW\_ERROR** 错误，因为 **100000** 超出了短整型值可容纳的数字范围。

从	始终	永不	与值相关
BINARY 或 VARBINARY	CHAR 或 VARCHAR BINARY LONG BINARY BIT TINYINT SHORT INT SIGNED SHORT LONG INT SIGNED LONG BIGINT SIGNED BIG	LONG VARCHAR REAL TIME TIMESTAMP DOUBLE DATE	NUMERIC UID <sup>1</sup>

从	始终	永不	与值相关
LONG BINARY	BINARY LONG BINARY	BIT CHAR 或 VARCHAR LONG VARCHAR TINYINT SHORT INT SIGNED SHORT LONG INT SIGNED LONG BIGINT SIGNED BIG REAL DOUBLE NUMERIC DATE TIME TIMESTAMP UID	N/A
BIT	CHAR 或 VARCHAR BINARY BIT TINYINT SHORT INT SIGNED SHORT LONG INT SIGNED LONG BIGINT REAL SIGNED BIG DOUBLE NUMERIC	LONG VARCHAR LONG BINARY DATE TIME TIMESTAMP UID	N/A



从	始终	永不	与值相关
CHAR 或 VARCHAR	BINARY 或 VARBINARY  CHAR 或 VARCHAR LONG VARCHAR	LONG BINARY	BIT  TINYINT  SIGNED SHORT  SHORT INT  LONG INT  SIGNED LONG  BIGINT  SIGNED BIG  DOUBLE  NUMERIC  REAL  DATE  TIME  TIMESTAMP  UID

从	始终	永不	与值相关
LONG VARCHAR	CHAR 或 VARCHAR LONG VARCHAR	BINARY 或 VARBINARY LONG BINARY BIT TINYINT SHORT INT SIGNED SHORT LONG INT SIGNED LONG BIGINT SIGNED BIG REAL NUMERIC DATE TIME TIMESTAMP DOUBLE UID	
TINYINT	BINARY 或 VARBINARY CHAR 或 VARCHAR TINYINT SHORT INT SIGNED SHORT LONG INT SIGNED LONG BIGINT SIGNED BIG REAL DOUBLE NUMERIC	LONG VARCHAR LONG BINARY DATE TIME TIMESTAMP UID	

从	始终	永不	与值相关
SHORT INT	BINARY 或 VARBINARY CHAR 或 VARCHAR SHORT INT LONG INT SIGNED LONG BIGINT SIGNED BIG REAL DOUBLE NUMERIC	LONG VARCHAR LONG BINARY DATE TIME TIMESTAMP UID	BIT TINYINT SIGNED SHORT
SIGNED SHORT	BINARY 或 VARBINARY CHAR 或 VARCHAR SIGNED SHORT SIGNED LONG SIGNED BIG REAL DOUBLE NUMERIC	LONG VARCHAR LONG BINARY DATE TIME TIMESTAMP UID	SHORT INT LONG INT BIGINT BIT TINYINT
LONG INT	BINARY 或 VARBINARY CHAR 或 VARCHAR LONG INT BIGINT SIGNED BIG REAL DOUBLE NUMERIC	LONG VARCHAR LONG BINARY DATE TIME TIMESTAMP UID	BIT TINYINT SHORT INT SIGNED SHORT SIGNED LONG

从	始终	永不	与值相关
SIGNED LONG	BINARY 或 VARBINARY CHAR 或 VARCHAR SIGNED LONG SIGNED BIG REAL DOUBLE NUMERIC DATE TIMESTAMP	LONG VARCHAR LONG BINARY TIME UID	BIT TINYINT SHORT INT SIGNED SHORT LONG INT BIGINT
BIGINT	BINARY 或 VARBINARY CHAR 或 VARCHAR BIGINT REAL DOUBLE NUMERIC	LONG VARCHAR LONG BINARY DATE TIME TIMESTAMP UID	BIT TINYINT SHORT INT SIGNED SHORT LONG INT SIGNED LONG SIGNED BIG
SIGNED BIG	BINARY 或 VARBINARY CHAR 或 VARCHAR SIGNED BIG REAL DOUBLE NUMERIC DATE TIMESTAMP	LONG VARCHAR LONG BINARY TIME UID	BIT TINYINT SHORT INT SIGNED SHORT LONG INT SIGNED LONG BIGINT

从	始终	永不	与值相关
REAL	CHAR 或 VARCHAR REAL DOUBLE NUMERIC	LONG VARCHAR BINARY 或 VARBINARY LONG BINARY DATE TIME TIMESTAMP UID	BIT TINYINT SHORT INT SIGNED SHORT LONG INT SIGNED LONG BIGINT SIGNED BIG
DOUBLE	CHAR 或 VARCHAR DOUBLE NUMERIC	LONG VARCHAR BINARY 或 VARBINARY LONG BINARY DATE TIME TIMESTAMP UID	BIT TINYINT SHORT INT SIGNED SHORT LONG INT SIGNED LONG BIGINT SIGNED BIG REAL
NUMERIC	CHAR 或 VARCHAR REAL NUMERIC DOUBLE	LONG VARCHAR LONG BINARY DATE TIME TIMESTAMP UID	BINARY 或 VARBINARY <sup>2</sup> BIT SHORT INT SIGNED SHORT LONG INT SIGNED LONG BIGINT SIGNED BIG TINYINT

从	始终	永不	与值相关
DATE	CHAR 或 VARCHAR SIGNED LONG SIGNED BIG DATE TIMESTAMP	LONG VARCHAR LONG BINARY BIT TINYINT SHORT INT SIGNED SHORT LONG INT BIGINT REAL DOUBLE NUMERIC TIME BINARY 或 VARBINARY UID	
TIME	CHAR 或 VARCHAR TIME TIMESTAMP	LONG VARCHAR LONG BINARY BIT TINYINT SHORT INT SIGNED SHORT LONG INT SIGNED LONG BIGINT SIGNED BIG REAL DOUBLE NUMERIC DATE BINARY 或 VARBINARY UID	

从	始终	永不	与值相关
TIMESTAMP	CHAR 或 VARCHAR SIGNED LONG SIGNED BIG DATE TIME TIMESTAMP	LONG VARCHAR LONG BINARY BIT TINYINT SHORT INT SIGNED SHORT LONG INT BIGINT REAL DOUBLE NUMERIC BINARY 或 VARBINARY UID	
UID	CHAR 或 VARCHAR UID	LONG VARCHAR LONG BINARY BIT TINYINT SHORT INT SIGNED SHORT LONG INT SIGNED LONG BIGINT SIGNED BIG REAL DOUBLE NUMERIC DATE TIME TIMESTAMP	BINARY 或 VARBINARY <sup>1</sup>

<sup>1</sup> BINARY 值的长度必须为 16 字节，以便与 UUID 兼容。

<sup>2</sup> 仅当源 NUMERIC 值能够转换为 BIGINT 时才适用。



## UltraLite 中的表达式

通常采用列引用的形式，通过将数据与运算符或函数进行组合构成表达式。

### 语法

```
expression:
  case-expression
  | constant
  | [correlation-name.]column-name
  | - expression
  | expression operator expression
  | ( expression )
  | function-name ( expression, ... )
  | if-expression
  | special value
  | input-parameter
```

### 参数

```
case-expression:
CASE expression
WHEN expression
THEN expression,...
[ ELSE expression ]
END

alternative form of case-expression:
CASE
WHEN search-condition
THEN expression,...
[ ELSE expression ]
END

constant:
integer | number | string | host-variable

special-value:
CURRENT { DATE | TIME | TIMESTAMP }
| NULL
| SQLCODE
| SQLSTATE

if-expression:
IF condition
THEN expression
[ ELSE expression ]
ENDIF

input-parameter:
{ ? | :name [ : indicator-name ] }

operator:
{ + | - | * | / | || | % }
```

**另请参见**

- “表达式中的常量”一节第 304 页
- “UltraLite 中的特殊值”一节第 286 页
- “表达式中的列名称”一节第 304 页
- “UltraLite SQL 函数”第 323 页
- “表达式中的子查询”一节第 307 页
- “UltraLite 中的搜索条件”一节第 309 页
- “UltraLite 中的数据类型”一节第 290 页
- “CASE 表达式”一节第 305 页
- “输入参数”一节第 308 页

## 表达式中的常量

在 UltraLite 中，常量是数字或字符串文字。

**语法**

```
'constant'
```

**用法**

字符串常量用单引号 (') 引起来。

在一行中使用两个单引号 (") 来表示字符串中的撇号。

**另请参见**

- “转义序列”一节 《SQL Anywhere 服务器 - SQL 参考》

**示例**

要使用所有格短语，请按如下所示键入字符串文字：

```
'John''s database'
```

## 表达式中的列名称

表达式中的标识符。

**语法**

```
correlation-name.column-name
```

**注释**

列名称跟在可选相关名（通常是表名）的后面。

如果列名称是关键字或含有字母、数字和下划线以外的其它字符，必须用引号 (") 将它引起来。例如，以下是有效的列名称：

```
Employees.Name  
address
```

```
"date hired"  
"salary"."date paid"
```

### 另请参见

- “UltraLite FROM 子句” 一节第 449 页

## IF 表达式

设置搜索条件以返回数据的特定子集。

### 语法 1

```
IF search-condition  
THEN expression1  
[ ELSE expression2 ]  
ENDIF
```

### 注释

出于兼容性考虑，此表达式可用 ENDIF 或 END IF 结尾。

该表达式的返回方式如下：

- 如果 *search-condition* 为 TRUE，则 IF 表达式返回 *expression1*。
- 如果 *search-condition* 为 FALSE 并且指定了 ELSE 子句，则 IF 表达式返回 *expression2*。
- 如果 *search-condition* 为 FALSE，且无 *expression2*，则 IF 表达式返回 NULL。
- 如果 *search-condition* 为 UNKNOWN，则 IF 表达式返回 NULL。

### 另请参见

- “NULL 值” 一节 《SQL Anywhere 服务器 - SQL 参考》
- “搜索条件” 一节 《SQL Anywhere 服务器 - SQL 参考》

## CASE 表达式

提供条件 SQL 表达式。

### 语法 1

```
CASE expression1  
WHEN expression2 THEN expression3, ...  
[ ELSE expression4 ]  
END
```

```
SELECT id,  
       ( CASE name  
         WHEN 'Tee Shirt' THEN 'Shirt'  
         WHEN 'Sweatshirt' THEN 'Shirt'  
         WHEN 'Baseball Cap' THEN 'Hat'  
         ELSE 'Unknown'
```

```
        END ) as Type
FROM Product;
```

## 语法 2

```
CASE
WHEN search-condition
THEN expression1, ...
[ ELSE expression2 ]
END
```

## 注释

出于兼容性考虑，此表达式可用 ENDCASE 或 END CASE 结尾。

Case 表达式可以在能够使用正则表达式的任何地方使用。

**语法 1** 如果 CASE 关键字后的表达式等于第一个 WHEN 关键字后的表达式，则返回相关联的 THEN 关键字后的表达式。否则，返回 ELSE 关键字后的表达式（如果已指定）。

例如，以下代码用 CASE 表达式作为 SELECT 语句的第二个子句。它从 Product 表中选择一行，其中名称列具有值 Sweatshirt。

**语法 2** 如果第一个 WHEN 关键字后的搜索条件为 TRUE，则返回相关联的 THEN 关键字后的表达式。否则，返回 ELSE 子句后的表达式（如果已指定）。

**用于简写 CASE 表达式的 NULLIF 函数** NULLIF 函数提供了一种方法来以短格式编写某些 CASE 语句。NULLIF 的语法如下：

```
NULLIF ( expression-1, expression-2 )
```

NULLIF 比较两个表达式的值。如果第一个表达式的值与第二个表达式的值相等，NULLIF 返回 NULL。如果它们不相等，NULLIF 则返回第一个表达式。

## 示例

以下语句使用 CASE 表达式作为 SELECT 语句的第三子句，以将字符串与搜索条件相关联。如果名称列的值为 **Tee Shirt**，则查询返回 **Sale**。如果名称列的值不是 **Tee Shirt** 且数量多于 50，则返回 **Big Sale**。但是，对于其它所有情况，查询返回 **Regular price**。

```
SELECT id, name,
       ( CASE
         WHEN name='Tee Shirt' THEN 'Sale'
         WHEN quantity >= 50 THEN 'Big Sale'
         ELSE 'Regular price'
         END ) as Type
FROM Product;
```

## 集合表达式

执行 UltraLite 运行时不提供的集合计算。

## 语法

```
SUM( expression )
```

## 注释

集合表达式从一定范围内的行计算单个值。

集合表达式中可以使用集合函数，或者其中一个或多个操作数为集合表达式。

当 SELECT 语句没有 GROUP BY 子句时，*select-list* 中的表达式要么必须包含所有的集合表达式，要么不包含集合表达式。当 SELECT 语句有 GROUP BY 子句时，*select-list* 中的所有非集合表达式都必须出现在 GROUP BY 列表中。

## 示例

例如，以下查询计算雇员表中雇员的工资总额。在此查询中，SUM( salary ) 就是集合表达式：

```
SELECT SUM( salary )
FROM employee;
```

## 表达式中的子查询

嵌套在另一个 SELECT 语句内部的 SELECT 语句。

## 语法

子查询的结构类似于常规查询。

## 注释

在 UltraLite 中，只能在以下情况下使用子查询引用：

- 作为 FROM 子句中的表表达式。这种形式的表表达式（也称为 **derived tables**）必须有一个派生表名和在其中读取 SELECT 列表中值的列名。
- 为 EXISTS、ANY、ALL 和 IN 搜索条件提供值。

可以编写子查询，其中含有对在子查询之前（子查询的左侧）指定的名称的引用（有时称为对左侧的外部引用）。但是，无法对子查询内的项进行引用（有时被称为内部引用）。

## 另请参见

- [“UltraLite SELECT 语句”一节第 457 页](#)
- [“使用子查询”《SQL Anywhere 服务器 - SQL 的用法》](#)
- [“UltraLite 中的搜索条件”一节第 309 页](#)

## 示例

以下子查询用于列出库存量较低（即少于 20）的项的全部产品 ID。

```
FROM SalesOrderItems
( SELECT ID
  FROM Products
  WHERE Quantity < 20 );
```

## 输入参数

充当占位符，以允许最终用户向预准备语句提供值。这些用户提供的值随后用于执行语句。

### 语法

```
{ ? | :name [ :indicator-name ] }
```

### 注释

在表达式中使用占位符 ? 或指定的形式。在能够使用列名称或常量的任何情况下，都能够使用输入参数。

用于向语句提供值的精确机制取决于创建 UltraLite 客户端所使用的 API。

**使用指定的形式** 输入参数的指定形式具有特定含义。通常，*name* 总是用于指定提供实际值的多个位置。

仅对于嵌入式 SQL 应用程序，*indicator-name* 会提供在其中放置空指示符的变量。如果将指定形式用于其它组件，将会忽略 *indicator-name*。

**推断数据类型** 当从以下模式之一预准备语句时，会推断输入参数的数据类型：

- **CAST ( ? AS type )**

在这种情况下，*type* 为数据库类型说明，如 CHAR(32)。

- 二进制运算符的一个操作数正好是输入参数。会将类型推断为操作数的类型。

如果无法推断类型，UltraLite 会生成错误。例如：

- **-?:** 操作数是一元的。
- **? + ?:** 两个操作数都是输入参数。

### 另请参见

- “使用主机变量”一节 《UltraLite - C 及 C++ 编程》
- “准备语句”一节 《SQL Anywhere 服务器 - 编程》
- UltraLite C/C++: “数据操作：Insert、Delete 和 Update”一节 《UltraLite - C 及 C++ 编程》
- UltraLite.NET: “数据操作：INSERT、UPDATE 和 DELETE”一节 《UltraLite - .NET 编程》
- UltraLite for M-Business: “数据操作：INSERT、UPDATE 和 DELETE”一节 《UltraLite - M-Business Anywhere 编程》

### 示例

以下嵌入式 SQL 语句具有两个输入参数：

```
INSERT INTO MyTable VALUES ( :v1, :v2, :v1);
```

v1 的第一个实例向语句中的 v2 和 v1 位置提供它的值。

## UltraLite 中的搜索条件

指定 WHERE 子句、HAVING 子句、连接中 ON 短语或 IF 表达式的搜索条件。搜索条件也称作谓  
语。

### 语法

```

search-condition:
  expression compare expression
| expression IS [ NOT ] { NULL | TRUE | FALSE | UNKNOWN }
| expression [ NOT ] BETWEEN expression AND expression
| expression [ NOT ] IN ( expression, ... )
| expression [ NOT ] IN ( subquery )
| expression [ NOT ] { ANY | ALL } ( subquery )
| expression [ NOT ] EXISTS ( subquery )
| NOT search-condition
| search-condition AND search-condition
| search-condition OR search-condition
| ( search-condition )

```

### 参数

```

compare:
= | > | < | >= | <= | <> | != | !< | !>

```

### 注释

在 UltraLite 中，搜索条件可出现在以下各项中：

- WHERE 子句
- HAVING 子句
- ON 短语
- SQL 查询

在 SELECT 语句的 FROM 子句中，或在表达式（如 IF 或 CASE）中，搜索条件可用于从表中选择行的子集以选择特定值。在 UltraLite 中，每个条件的值都处于以下三种状态中的一种：TRUE、FALSE 或 UNKNOWN。组合时，这些状态称为 **three-valued logic**。如果所比较的值中某一个为 NULL，则比较结果为 UNKNOWN。仅当条件的结果为 TRUE 时，才满足搜索条件。

UltraLite 支持的不同类型的搜索条件包括：

- ALL 条件
- ANY 条件
- BETWEEN 条件
- EXISTS 条件
- IN 条件

这些条件将在随后的单独章节中进行论述。

#### 注意

子查询构成了一类很重要的表达式，该表达式可用在很多搜索条件中。

**另请参见**

- “比较运算符”一节第 310 页
- “三值逻辑”一节《SQL Anywhere 服务器 - SQL 参考》
- “表达式中的子查询”一节第 307 页

**比较运算符**

允许在搜索条件中对两个或多个表达式进行比较的任何运算符。

**语法**

*expression operator expression*

**参数**

运算符	解释
=	等于
[ NOT ] LIKE	文本比较，可能使用正则表达式
>	大于
<	小于
>=	大于或等于
<=	小于或等于
!=	不等于
<>	不等于
!>	不大于
!<	不小于

**注释**

**比较日期** 比较日期时，< 表示日期较早，> 表示日期较晚。

**比较 LONG VARCHAR 或 LONG BINARY 值** UltraLite 不支持使用 LONG VARCHAR 或 LONG BINARY 值进行比较。

**区分大小写** 在 UltraLite 中，执行比较时不仅要注意进行操作的数据库，还需注意大小写。缺省情况下，创建 UltraLite 数据库时不区分大小写。

**NOT 运算符** NOT 运算符对表达式取非。



## 另请参见

- “逻辑运算符”一节第 311 页
- “UltraLite 中的搜索条件”一节第 309 页

## 示例

以下两个查询中的任何一个都将查找所有价格为 \$10（或不到）的 T 恤衫和棒球帽。但是，请注意否定逻辑运算符 (NOT) 和否定比较运算符 (!>) 之间的差别。

```
SELECT ID, Name, Quantity
FROM Products
WHERE (name = 'Tee Shirt' OR name = 'BaseBall Cap')
AND NOT UnitPrice > 10;
```

```
SELECT ID, Name, Quantity
FROM Products
WHERE (name = 'Tee Shirt' OR name = 'BaseBall Cap')
AND UnitPrice !> 10;
```

## 逻辑运算符

执行以下操作之一：

- 比较条件 (AND、OR 和 NOT)。
- 测试表达式 (IS) 的真值或 NULL 值特性。

### 语法 1

*condition1* *logical-operator* *condition2*

### 语法 2

**NOT** *condition*

### 语法 3

*expression* **IS** [ **NOT** ] { *truth-value* | **NULL** }

## 注释

在 SELECT 语句的 FROM 子句中，或在表达式（如 IF 或 CASE）中，搜索条件可用于从表中选择行的子集以选择特定值。在 UltraLite 中，每个条件的值都处于以下三种状态中的一种：TRUE、FALSE 或 UNKNOWN。组合时，这些状态称为 **three-valued logic**。如果所比较的值中某一个为 NULL，则比较结果为 UNKNOWN。仅当条件的结果为 TRUE 时，才满足搜索条件。

**AND** 如果两个条件都为 TRUE，则组合条件为 TRUE；如果其中某一条件为 FALSE，则组合条件为 FALSE；其它情况下为 UNKNOWN。

*condition1* **OR** *condition2*

**OR** 如果其中某一条件为 TRUE，则组合条件为 TRUE；如果两个条件都为 FALSE，则组合条件为 FALSE；其它情况下为 UNKNOWN。

**NOT** 如果 *condition* 为 FALSE，则 NOT 条件为 TRUE；如果 *condition* 为 TRUE，则 NOT 条件为 FALSE；如果 *condition* 为 UNKNOWN，则 NOT 条件为 UNKNOWN。

**IS** 如果 *expression* 的值为提供的 *truth-value*（必须是 TRUE、FALSE 或 UNKNOWN 其中之一），则条件为 TRUE。否则，条件值为 FALSE。

#### 另请参见

- “三值逻辑”一节 《SQL Anywhere 服务器 - SQL 参考》
- “比较运算符”一节第 310 页
- “UltraLite 中的搜索条件”一节第 309 页

#### 示例

如果列包含 NULL 值，则满足 IS NULL 条件。如果使用 IS NOT NULL 运算符，则当列包含不是 NULL 的值时满足该条件。此示例显示一个 IS NULL 条件：WHERE paid\_date IS NULL。

## ALL 条件

将 ALL 条件与比较运算符联合使用，可以将单个值与子查询生成的数据值进行比较。

#### 语法

```
expression compare [ NOT ] ALL ( subquery )
```

#### 参数

*compare*:  
= | > | < | >= | <= | <> | != | !< | !>

#### 注释

UltraLite 使用指定的比较运算符将测试值与结果集中的每一数据值进行比较。如果所有比较运算生成的结果都为 TRUE，则 ALL 测试返回 TRUE。

#### 另请参见

- “子查询和 ALL 测试”一节 《SQL Anywhere 服务器 - SQL 的用法》
- “比较运算符”一节第 310 页

#### 示例

查找在订单 #2001 的所有产品都已发运之后所下的那些订单的订单 ID 和客户 ID。

```
SELECT ID, CustomerID
FROM SalesOrders
WHERE OrderDate > ALL (
  SELECT ShipDate
  FROM SalesOrderItems
  WHERE ID=2001);
```

## ANY 条件

将 ANY 条件与比较运算符联合使用，可以将单个值与子查询生成的一系列数据值进行比较。

### 语法 1

```
expression compare [ NOT ] ANY ( subquery )
```

### 语法 2

```
expression = ANY ( subquery )
```

### 参数

```
compare:  
= | > | < | >= | <= | <> | != | !< | !>
```

### 注释

UltraLite 使用指定的比较运算符将测试值与列中的每一数据值进行比较。如果任何比较运算生成的结果为 TRUE，则 ANY 测试将返回 TRUE。

**语法 1** 如果 *expression* 等于子查询结果中的任何一个值，则为 TRUE；如果表达式不为 NULL 而且不等于子查询的任何返回值，则为 FALSE。如果 *expression* 的值为 NULL，则 ANY 条件为 UNKNOWN，除非子查询的结果中没有行，在这种情况下，该条件始终为 FALSE。

### 另请参见

- “子查询和 ANY 测试”一节 《SQL Anywhere 服务器 - SQL 的用法》
- “比较运算符”一节第 310 页

### 示例

查找在订单 #2005 的第一批产品已发出之后所发的那些订单的订单 ID 和客户 ID。

```
SELECT ID, CustomerID  
FROM SalesOrders  
WHERE OrderDate > ANY (  
  SELECT ShipDate  
  FROM SalesOrderItems  
  WHERE ID=2005);
```

## BETWEEN 条件

指定包含的范围，将搜索该范围的上限值、下限值以及介于这两个值之间的值。

### 语法

```
expression [ NOT ] BETWEEN start-expression AND end-expression
```

### 注释

BETWEEN 条件的值可以是 TRUE、FALSE 或 UNKNOWN。无 NOT 关键字时，如果 *expression* 介于 *start-expression* 和 *end-expression* 之间，则条件的值为 TRUE。关键字 NOT 使条件的含义相反，但保留 UNKNOWN 不变。

BETWEEN 条件相当于两个不等式的组合:

```
[ NOT ] ( expression >= start-expression  
         AND expression <= end-expression )
```

### 示例

列出价格低于 \$10 或高于 \$15 的所有产品。

```
SELECT Name, UnitPrice  
FROM Products  
WHERE UnitPrice NOT BETWEEN 10 AND 15;
```

## EXISTS 条件

检查子查询是否生成一些查询结果行。

### 语法

```
[ NOT ] EXISTS ( subquery )
```

### 注释

如果子查询结果至少包含一行，则 EXISTS 条件为 TRUE；如果子查询结果一行也不包含，则条件为 FALSE。EXISTS 条件不能为 UNKNOWN。

您可以使用 NOT EXISTS 形式反转 EXISTS 条件的逻辑。在此情况下，如果子查询没有生成任何行，则该测试返回 TRUE，否则返回 FALSE。

### 示例

列出在 2001 年 7 月 13 日以后下订单的客户。

```
SELECT GivenName, Surname  
FROM Customers  
WHERE EXISTS (  
  SELECT *  
  FROM SalesOrders  
  WHERE (OrderDate > '2001-07-13') AND  
         (Customers.ID = SalesOrders.CustomerID));
```

## IN 条件

通过从主查询中搜索一个值同时在子查询中搜索另一个值来检查所属关系。

### 语法

```
expression [ NOT ] IN  
{ ( subquery ) | ( value-expr, ... ) }
```

### 参数

*value-expr* 是具有单值的表达式，它的值可以是字符串、数字、日期或任何其它 SQL 数据类型。

## 注释

没有 NOT 关键字时, IN 条件的值按照以下规则计算:

- 如果 *expression* 不为 NULL 并且至少等于其中一个值, 则值为 TRUE。
- 如果 *expression* 为 NULL 而值列表非空, 或者如果至少其中一个值为 NULL 且 *expression* 不等于任何其它值, 则值为 UNKNOWN。
- 如果 *expression* 为 NULL 且 *subquery* 不返回任何值, 或者如果 *expression* 不为 NULL, 所有值均不为 NULL, 并且 *expression* 不等于其中任何一个值, 则值为 FALSE。

您可以使用 NOT IN 形式反转 IN 条件的逻辑。

以下搜索条件 *expression* **IN** (*values*) 与搜索条件 *expression* = **ANY** (*values*) 等同。以下搜索条件 *expression* **NOT IN** (*values*) 与搜索条件 *expression*  $\not\in$  **ALL** (*values*) 等同。

## 示例

选择居住在以下加拿大省份的客户的公司名和州: Ontario、Manitoba 和 Quebec。

```
SELECT CompanyName , Province
FROM Customers
WHERE State IN( 'ON', 'MB', 'PQ');
```

## UltraLite 中的运算符

运算符用于计算值，而这些值又可能用作更高级别表达式中的操作数。

UltraLite SQL 支持以下类型的运算符：

- 比较运算符使用一个（一元）或两个（二元）比较操作数计算并返回结果。比较会得出三种常见的逻辑值：**true**、**false** 和 **unknown**。
- 算术运算符会计算并返回全部浮点数、十进制数和整数的结果集。
- 字符串运算符将两个字符串值连接在一起。例如，"my" + "string" 返回字符串 "my string"。
- 位运算符计算特定位并在整数的内部表示中打开或关闭它。
- 逻辑运算符计算搜索条件。逻辑计算会得出三种常见的逻辑值：**true**、**false** 和 **unknown**。

正常的运算优先级顺序仍适用。

### 另请参见

- “运算符优先级”一节第 317 页
- “比较运算符”一节第 310 页
- “算术运算符”一节第 316 页
- “字符串运算符”一节第 317 页
- “位运算符”一节第 317 页
- “逻辑运算符”一节第 311 页

## 算术运算符

算术运算符能够使您执行计算。

**expression + expression** 加法。如果其中某一表达式为 NULL，则结果为 NULL。

**expression - expression** 减法。如果其中某一表达式为 NULL，则结果为 NULL。

**- expression** 取非。如果表达式为 NULL，则结果为 NULL。

**expression \* expression** 乘法。如果其中某一表达式为 NULL，则结果为 NULL。

**expression / expression** 除法。如果其中某一表达式为 NULL，或第二个表达式为 0，则结果为 NULL。

**expression % expression** 模运算结果为两个整数相除后的整数余数。例如， $21 \% 11 = 10$ ，因为 21 除以 11 等于 1，余数为 10。如果其中某一表达式为 NULL，则结果为 NULL。

### 另请参见

- “算术运算”一节 《SQL Anywhere 服务器 - SQL 的用法》

## 字符串运算符

字符串运算符能够连接字符串—LONGVARCHAR 和 LONGBINARY 数据类型除外。

**expression || expression** 字符串连接（两个竖线）。如果其中某一字符串为 NULL，则连接时将其视为空字符串。

**expression + expression** 另一种字符串连接运算方式。使用 + 连接运算符时，必须确保操作数已显式设置为字符数据类型，而不是依赖隐式的数据转换。

例如，以下查询返回整数值 **579**：

```
SELECT 123 + 456;
```

而以下查询则返回字符串 **123456**：

```
SELECT '123' + '456';
```

可以使用 CAST 或 CONVERT 函数来显式地转换数据类型。

## 位运算符

位运算符在两个表达式间执行位操作。可在 UltraLite 中将以下运算符用于整型数据类型。

运算符	说明
&	逐位与
	逐位或
^	逐位异或
~	逐位非

位运算符 &、| 和 ~ 与逻辑运算符 AND、OR 和 NOT 不能互换。位运算符使用值的位表示形式运算整数值。

### 示例

以下语句选择设置了指定位的行。

```
SELECT *
FROM tableA
WHERE (options & 0x0101) <> 0;
```

## 运算符优先级

表达式的运算符优先级如下。圆括号中的表达式先进行运算，然后是乘除，继而加减。字符串连接在加减之后进行。列表顶部的运算符先于底部的运算符进行运算。

**提示**

显式设置 UltraLite 中的运算顺序，而不是依赖运算符优先级。这意味着，在一个表达式中使用一个以上的运算符时，应该使用圆括号明确地为运算排序。

1. 名称、函数、常量、IF 表达式、CASE 表达式
2. ()
3. 一元运算符（需要单操作数的运算符）：+, -
4. ~
5. &, |, ^
6. \*, /, %
7. +, -
8. ||
9. 比较：>, <, <>, !=, <=, >=, [ NOT ] BETWEEN, [ NOT ] IN, [ NOT ] LIKE
10. 比较：IS [NOT] TRUE, FALSE, UNKNOWN
11. NOT
12. AND
13. OR



## UltraLite 中的变量

不能在 UltraLite 应用程序中使用 SQL 变量（包括全局变量）。

## UltraLite 中的执行计划

UltraLite 执行计划显示在执行查询时如何访问表和索引。UltraLite 包括一个 **query optimizer**。优化程序是 UltraLite 运行时的一个内部组件，尝试生成高效的查询计划。优化程序会尝试避免使用临时表来存储中间结果，并尝试确保在查询连接两个表时仅访问表的相关子集。

### 替换优化程序

优化程序总是尽可能地找出最高效的访问计划，但这一目标却不能得到保证——尤其是对于存在多种可能性的复杂查询。在极个别情况下，您可以通过向查询添加 `OPTION (FORCE ORDER)` 子句替换优化程序选择的表顺序，该子句可以强制 UltraLite 以表在查询中的出现顺序对表进行访问。不建议将此选项用于通用目的。如果性能较低，通常较好的方法是创建适当的索引来加快执行速度。

#### 性能提示

如果不打算通过查询更新数据，则应该在查询中指定 `FOR READ ONLY` 子句。这个子句可能会产生更好的性能。请参见“[UltraLite SELECT 语句](#)”一节第 457 页。

## 查看执行计划时

在 Interactive SQL 中查看查执行计划时，您需要了解：

- 将用于返回结果的索引。索引扫描对象包含表的名称和正在该表上使用的索引。
- 是否使用临时表来返回结果。临时表会写入 UltraLite 临时文件。请参见“[UltraLite 临时文件](#)”一节第 10 页。
- 表的连接顺序。此信息能够使您确定如何影响性能。
- 查询运行速度慢的原因或者确保查询不会运行得很慢。

## 查看 UltraLite 执行计划

作为开发过程的一种辅助手段，可以使用 Interactive SQL 显示总结预准备语句的执行方式的 UltraLite 计划。文本计划显示在 Interactive SQL 计划查看器中。

在 UltraLite 中，执行计划必须是计划的简要文本摘要。不支持其它计划类型。但是，简要计划能够使您快速地比较计划，因为信息汇总在一行上。

### ◆ 在计划查看器中查看执行计划

1. 选择 [工具] » [计划查看器]。
2. 在 [SQL 语句] 窗格中，键入查询。
3. 单击 [获取计划] 为指定的 SQL 语句生成计划。

## 示例

该文本计划出现在计划查看器的底部窗格中。

请看以下语句：

```
SELECT I.inv_no, I.name, T.quantity, T.prod_no
FROM Invoice I, Transactions T
WHERE I.inv_no = T.inv_no;
```

此语句可能会生成以下计划：

```
join[scan(Invoice,primary),index-scan(Transactions,secondary)]
```

此计划指出连接操作需执行以下操作：读取 Invoice 表（根据指定的主索引）中的所有行。然后使用指定的次级索引从 Transaction 表中仅读取 inv\_no 列匹配的行。

## 另请参见

- “Interactive SQL 实用程序 (dbisql)” 一节 《SQL Anywhere 服务器 - 数据库管理》
- “读取 UltraLite 执行计划” 一节第 321 页

## 读取 UltraLite 执行计划

由于 UltraLite 简要计划是如何访问查询的文本概括，您需要了解表的连接或扫描操作是如何实现的。

- **对于扫描操作** 用单个操作数表示，只适用于单个表并且使用索引。表名和索引名以圆括号 ((),) 方式显示，接在操作名后。
- **对于其它操作** 用一个或多个操作数表示，还可以是自身中的计划和自身的计划。在 UltraLite 中，这些操作数是由逗号分隔的列表，包含在方括号 ([ ]) 内。

## 操作列表

下表列出了 UltraLite 支持的操作。

操作	说明
<b>count(*)</b>	计算表中的行数。
<b>distinct[ plan ]</b>	执行查询的 DISTINCT 方面，以比较并消除重复行。它可用于在基础计划对行进行排序时消除重复的连续行。如果两个连续的行匹配，只会将第一行添加到结果集中。
<b>dummy</b>	不执行任何操作。它只在两种情况中出现： <ul style="list-style-type: none"> <li>● 当在 FROM 子句中指定 DUMMY 时。</li> <li>● 当查询中缺少 FROM 子句时。</li> </ul>

操作	说明
<b>filter</b> [ <i>plan</i> ]	执行由基础计划提供的各行的搜索条件。只有值为 true 的行才会作为结果集的一部分发送。
<b>group-by</b> [ <i>plan</i> ]	创建 GROUP BY 结果的集合，以便对分组数据的多行进行排序。行按照出现的顺序列出，并且通过比较连续行进行分组。
<b>group-single</b> [ <i>plan</i> ]	创建 GROUP BY 结果的集合，但是只在知道将会返回单行时。
<b>+keyset</b> [ <i>plan</i> ]	记录用于创建临时表行的那些行，以便 UltraLite 可以更新原始行。如果不想更新那些行，那么应在查询中使用 FOR READ ONLY 子句来消除此操作。
<b>index-scan</b> ( <i>table-name</i> , <i>index-name</i> )	只读取表的一部分；索引用于查找起始行。
<b>join</b> [ <i>plan</i> , <i>plan</i> ]	在两个计划之间进行内连接。
<b>lojoin</b> [ <i>plan</i> , <i>plan</i> ]	在两个计划之间进行左外连接。
<b>like-scan</b> ( <i>table-name</i> , <i>index-name</i> )	只读表的一部分；索引用于通过模式匹配查找起始行。
<b>rowlimit</b> [ <i>plan</i> ]	对传播行执行行限制操作。行限制是通过 SELECT 语句的 TOP n 或 FIRST 子句来设置的。
<b>scan</b> ( <i>table-name</i> , <i>index-name</i> )	按照索引指示的顺序读取整个表。
<b>sub-query</b> [ <i>plan</i> ]	标记子查询的起始。
<b>temp</b> [ <i>plan</i> ]	<p>从基础计划中的行创建临时表。当必须对基础行进行排序而又找不到索引来完成此排序时，UltraLite 会使用临时表。</p> <p>可以添加一个索引来消除对临时表的需求。但是，使用的每个附加索引都会增加在应用索引的表插入或同步行所需的持续时间。</p>
<b>union-all</b> [ <i>plan</i> , ..., <i>plan</i> ]	对在基础计划中生成的行执行 UNION ALL 操作。

---

# UltraLite SQL 函数

## 目录

函数类型 .....	324
SQL 函数 (A-D) .....	330
SQL 函数 (E-O) .....	359
SQL 函数 (P-Z) .....	386

---

函数用于从数据库返回信息。在任何可以使用表达式的地方都可以使用函数。

除非在文档中另有指定，否则如果有任何参数为 NULL，函数都将返回 NULL。

函数使用的语约定与 SQL 语句使用的语约定相同。有关语约定的完整列表，请参见“[语约定](#)”一节《[SQL Anywhere 服务器 - SQL 参考](#)》。

## 函数类型

本节将按类型对可用的函数进行分组。

UltraLite 支持的函数是所介绍的 SQL Anywhere 函数的一部分，但有时存在一些区别。

**注意**

除非另外说明，任何接收 NULL 作为参数的函数都返回 NULL。

## UltraLite 集合函数

集合函数对数据库中一组行的数据进行汇总。这些组是使用 SELECT 语句的 GROUP BY 子句构成的。只允许在选择列表和 SELECT 语句的 HAVING 和 ORDER BY 子句中使用集合函数。

### 函数列表

以下是可用的集合函数：

- “AVG 函数 [Aggregate]” 一节第 335 页
- “COUNT 函数 [Aggregate]” 一节第 345 页
- “LIST 函数 [Aggregate]” 一节第 371 页
- “MAX 函数 [Aggregate]” 一节第 376 页
- “MIN 函数 [Aggregate]” 一节第 376 页
- “SUM 函数 [Aggregate]” 一节第 406 页

## UltraLite 数据类型转换函数

数据类型转换函数用于将参数从一种数据类型转换为另一种数据类型，或测试它们是否可以转换。

### 函数列表

以下是可用的数据类型转换函数：

- “CAST 函数 [Data type conversion]” 一节第 337 页
- “CONVERT 函数 [Data type conversion]” 一节第 342 页
- “HEXTOINT 函数 [Data type conversion]” 一节第 362 页
- “INTTOHEX 函数 [Data type conversion]” 一节第 366 页
- “ISDATE 函数 [Data type conversion]” 一节第 367 页

## UltraLite 日期和时间函数

日期和时间函数对日期和时间数据类型执行操作，或者返回日期或时间信息。

在本章中，**日期时间** 一词用来表示日期或时间，或者时间戳。具体的数据类型 DATETIME 表示为 DATETIME。

有关日期时间数据类型的详细信息，请参见“UltraLite 中的数据类型”一节第 290 页。

## 函数列表

以下是可用的日期和时间函数：

- “DATE 函数 [Date and time]” 一节第 347 页
- “DATEADD 函数 [Date and time]” 一节第 348 页
- “DATEDIFF 函数 [Date and time]” 一节第 349 页
- “DATEFORMAT 函数 [Date and time]” 一节第 350 页
- “DATENAME 函数 [Date and time]” 一节第 351 页
- “DATEPART 函数 [Date and time]” 一节第 351 页
- “DATETIME 函数 [Date and time]” 一节第 352 页
- “DAY 函数 [Date and time]” 一节第 353 页
- “DAYNAME 函数 [Date and time]” 一节第 353 页
- “DAYS 函数 [Date and time]” 一节第 354 页
- “DOW 函数 [Date and time]” 一节第 357 页
- “GETDATE 函数 [Date and time]” 一节第 361 页
- “HOUR 函数 [Date and time]” 一节第 363 页
- “HOURS 函数 [Date and time]” 一节第 364 页
- “MINUTE 函数 [Date and time]” 一节第 377 页
- “MINUTES 函数 [Date and time]” 一节第 378 页
- “MONTH 函数 [Date and time]” 一节第 381 页
- “MONTHNAME 函数 [Date and time]” 一节第 381 页
- “MONTHS 函数 [Date and time]” 一节第 382 页
- “NOW 函数 [Date and time]” 一节第 384 页
- “QUARTER 函数 [Date and time]” 一节第 388 页
- “SECOND 函数 [Date and time]” 一节第 395 页
- “SECONDS 函数 [Date and time]” 一节第 396 页
- “TODAY 函数 [Date and time]” 一节第 410 页
- “WEEKS 函数 [Date and time]” 一节第 417 页
- “YEAR 函数 [Date and time]” 一节第 418 页
- “YEARS 函数 [Date and time]” 一节第 419 页
- “YMD 函数 [Date and time]” 一节第 420 页

## 日期部分

许多日期函数都使用从**日期部分** 生成的日期。下表显示了允许的日期部分的值。

日期部分	缩写	值
年份	yy	1-9999
季度	qq	1-4

日期部分	缩写	值
Month	mm	1-12
Week	wk	1-54. 一周从星期日开始。
Day	dd	1-31
Dayofyear	dy	1-366
Weekday	dw	1-7 (星期日 = 1, ……., 星期六 = 7)
Hour	hh	0-23
Minute	mi	0-59
Second	ss	0-59
Millisecond	ms	0-999
Calyearofweek	cyr	整数。一周开始的年份。包含一年头几天的周可能在前一年已经开始，具体取决于这一年从星期几开始。从星期一到星期四开始的年没有属于前一年的天，但从星期五到星期日开始的年在这一年的第一个星期一开始第一周。
Calweekofyear	cwk	1-54. 一年内包含指定日期的周的顺序号。
Caldayofweek	cdw	1-7. (星期一 = 1, ……., 星期日 = 7)

## UltraLite 杂项函数

杂类函数对算术、字符串或日期/时间表达式（包括其它函数的返回值）执行运算。

### 函数列表

以下是可用的杂类函数：

- [“ARGN 函数 \[Miscellaneous\]” 一节第 331 页](#)
- [“COALESCE 函数 \[Miscellaneous\]” 一节第 341 页](#)
- [“EXPLANATION 函数 \[Miscellaneous\]” 一节第 359 页](#)
- [“GREATER 函数 \[Miscellaneous\]” 一节第 361 页](#)
- [“IFNULL 函数 \[Miscellaneous\]” 一节第 365 页](#)
- [“ISNULL 函数 \[Miscellaneous\]” 一节第 367 页](#)
- [“LESSER 函数 \[Miscellaneous\]” 一节第 370 页](#)
- [“NEWID 函数 \[Miscellaneous\]” 一节第 383 页](#)
- [“NULLIF 函数 \[Miscellaneous\]” 一节第 384 页](#)



## UltraLite 数字函数

数字函数执行数字数据类型的数学运算或者返回数字信息。

### 函数列表

以下是可用的数字函数：

- “ABS 函数 [Numeric]” 一节第 330 页
- “ACOS 函数 [Numeric]” 一节第 330 页
- “ASIN 函数 [Numeric]” 一节第 332 页
- “ATAN 函数 [Numeric]” 一节第 333 页
- “ATAN2 函数 [Numeric]” 一节第 334 页
- “CEILING 函数 [Numeric]” 一节第 338 页
- “COS 函数 [Numeric]” 一节第 344 页
- “COT 函数 [Numeric]” 一节第 345 页
- “DEGREES 函数 [Numeric]” 一节第 356 页
- “EXP 函数 [Numeric]” 一节第 359 页
- “FLOOR 函数 [Numeric]” 一节第 360 页
- “LOG 函数 [Numeric]” 一节第 373 页
- “LOG10 函数 [Numeric]” 一节第 373 页
- “MOD 函数 [Numeric]” 一节第 380 页
- “PI 函数 [Numeric]” 一节第 387 页
- “POWER 函数 [Numeric]” 一节第 387 页
- “RADIANS 函数 [Numeric]” 一节第 389 页
- “REMAINDER 函数 [Numeric]” 一节第 389 页
- “ROUND 函数 [Numeric]” 一节第 394 页
- “SIGN 函数 [Numeric]” 一节第 398 页
- “SIN 函数 [Numeric]” 一节第 399 页
- “SQRT 函数 [Numeric]” 一节第 401 页
- “TAN 函数 [Numeric]” 一节第 409 页
- “TRUNCNUM 函数 [Numeric]” 一节第 411 页

## UltraLite 字符串函数

字符串函数执行字符串的转换、抽取或处理操作，或者返回有关字符串的信息。

使用多字节字符集工作时，请仔细检查所使用的函数是否返回有关字符或字节的信息。

## 函数列表

以下是可用的字符串函数：

- “ASCII 函数 [String]” 一节第 332 页
- “BYTE\_LENGTH 函数 [String]” 一节第 335 页
- “BYTE\_SUBSTR 函数 [String]” 一节第 336 页
- “CHAR 函数 [String]” 一节第 339 页
- “CHARINDEX 函数 [String]” 一节第 339 页
- “CHAR\_LENGTH 函数 [String]” 一节第 340 页
- “DIFFERENCE 函数 [String]” 一节第 356 页
- “INSERTSTR 函数 [String]” 一节第 366 页
- “LCASE 函数 [String]” 一节第 368 页
- “LEFT 函数 [String]” 一节第 369 页
- “LENGTH 函数 [String]” 一节第 370 页
- “LOCATE 函数 [String]” 一节第 372 页
- “LOWER 函数 [String]” 一节第 374 页
- “LTRIM 函数 [String]” 一节第 375 页
- “PATINDEX 函数 [String]” 一节第 386 页
- “REPEAT 函数 [String]” 一节第 390 页
- “REPLACE 函数 [String]” 一节第 391 页
- “REPLICATE 函数 [String]” 一节第 392 页
- “RIGHT 函数 [String]” 一节第 393 页
- “RTRIM 函数 [String]” 一节第 394 页
- “SIMILAR 函数 [String]” 一节第 398 页
- “SOUNDEX 函数 [String]” 一节第 400 页
- “SPACE 函数 [String]” 一节第 401 页
- “STR 函数 [String]” 一节第 402 页
- “STRING 函数 [String]” 一节第 403 页
- “STRTOUUID 函数 [String]” 一节第 404 页
- “STUFF 函数 [String]” 一节第 404 页
- “SUBSTRING 函数 [String]” 一节第 405 页
- “TRIM 函数 [String]” 一节第 410 页
- “UCASE 函数 [String]” 一节第 413 页
- “UPPER 函数 [String]” 一节第 414 页
- “UIDTOSTR 函数 [String]” 一节第 416 页

## UltraLite 系统函数

系统函数用于返回系统信息。

---

## 函数列表

以下是 UltraLite 中可用的系统函数:

- “DB\_PROPERTY 函数 [System]” 一节第 355 页
- “ML\_GET\_SERVER\_NOTIFICATION [系统]” 一节第 379 页
- “SYNC\_PROFILE\_OPTION\_VALUE 函数 [系统]” 一节第 408 页

## SQL 函数 (A-D)

列出了每个函数，并在它旁边指出函数类型（数字、字符等等）。

有关指向给定类型所有函数的链接，请参见“函数类型”一节第 324 页。

### 另请参见

- “SQL 函数 (E-O)”一节 《SQL Anywhere 服务器 - SQL 参考》
- “SQL 函数 (P-Z)”一节 《SQL Anywhere 服务器 - SQL 参考》

## ABS 函数 [Numeric]

返回数字表达式的绝对值。

### 语法

**ABS**( *numeric-expression* )

### 参数

- **numeric-expression** 要返回其绝对值的数字。

### 返回值

数字表达式的绝对值。

数字表达式数据类型	返回值
INT	INT
FLOAT	FLOAT
DOUBLE	DOUBLE
NUMERIC	NUMERIC

### 标准和兼容性

- **SQL/2003** 核心 SQL 之外的 SQL 基础特性。

### 示例

以下语句返回值 66。

```
SELECT ABS ( -66 );
```

## ACOS 函数 [Numeric]

返回数字表达式的反余弦值（以弧度表示）。

**语法**

**ACOS**( *numeric-expression* )

**参数**

- **numeric-expression** 角度的余弦值。

**返回值**

DOUBLE

**注释**

此函数将其参数转换为 DOUBLE，并以双精度浮点执行计算。

**另请参见**

- [“ASIN 函数 \[Numeric\]” 一节第 332 页](#)
- [“ATAN 函数 \[Numeric\]” 一节第 333 页](#)
- [“ATAN2 函数 \[Numeric\]” 一节第 334 页](#)
- [“COS 函数 \[Numeric\]” 一节第 344 页](#)

**标准和兼容性**

- **SQL/2003** 服务商扩充。

**示例**

以下语句返回 0.52 的反余弦值。

```
SELECT ACOS ( 0.52 );
```

## ARGN 函数 [Miscellaneous]

从参数列表中返回所选参数。

**语法**

**ARGN**( *integer-expression*, *expression* [ , ... ] )

**参数**

- **integer-expression** 表达式列表中的参数位置。
- **expression** 任何传递给函数的数据类型的表达式。提供的所有表达式都必须属于相同的数据类型。

**返回值**

使用 *integer-expression* 的值作为 **n**，从其余参数列表中返回第 **n** 个参数（从 1 开始）。

**注释**

虽然表达式可以是任意数据类型，但它们必须具有相同的数据类型。整数表达式必须是从 1 到列表中的表达式数目，否则返回 NULL。多个表达式之间以逗号分隔。

### 标准和兼容性

- **SQL/2003** 服务商扩充。

### 示例

以下语句返回值 6。

```
SELECT ARGN( 6, 1,2,3,4,5,6 );
```

## ASCII 函数 [String]

返回字符串表达式中第一个字节的整数 ASCII 值。

### 语法

**ASCII**( *string-expression* )

### 参数

- **string-expression** 字符串。

### 返回值

SMALLINT

### 注释

如果字符串为空，则 ASCII 返回零。文字字符串必须用引号括起来。如果数据库字符集为多字节字符集，并且参数字符串的第一个字符包含一个以上的字节，则结果为 NULL。

### 另请参见

- “UltraLite 字符串函数” 一节第 327 页

### 标准和兼容性

- **SQL/2003** 服务商扩充。

### 示例

以下语句返回值 90。

```
SELECT ASCII( 'Z' );
```

## ASIN 函数 [Numeric]

返回一个数字的反正弦值（以弧度表示）。

### 语法

**ASIN**( *numeric-expression* )

### 参数

- **numeric-expression** 角度的正弦值。

### 返回值

DOUBLE

### 注释

SIN 和 ASIN 函数互为逆运算。

此函数将其参数转换为 DOUBLE，并以双精度浮点执行计算。

### 另请参见

- [“ACOS 函数 \[Numeric\]” 一节第 330 页](#)
- [“ATAN 函数 \[Numeric\]” 一节第 333 页](#)
- [“ATAN2 函数 \[Numeric\]” 一节第 334 页](#)
- [“SIN 函数 \[Numeric\]” 一节第 399 页](#)

### 标准和兼容性

- **SQL/2003** 服务商扩充。

### 示例

以下语句返回 0.52 的反正弦值。

```
SELECT ASIN( 0.52 );
```

## ATAN 函数 [Numeric]

返回一个数字的反正切值（以弧度表示）。

### 语法

**ATAN**( *numeric-expression* )

### 注释

ATAN 和 TAN 函数互为逆运算。

### 参数

- **numeric-expression** 角度的正切值。

### 返回值

DOUBLE

### 注释

此函数将其参数转换为 DOUBLE，并以双精度浮点执行计算。

**另请参见**

- [“ACOS 函数 \[Numeric\]” 一节第 330 页](#)
- [“ASIN 函数 \[Numeric\]” 一节第 332 页](#)
- [“ATAN2 函数 \[Numeric\]” 一节第 334 页](#)
- [“TAN 函数 \[Numeric\]” 一节第 409 页](#)

**标准和兼容性**

- **SQL/2003** 服务商扩充。

**示例**

以下语句返回 0.52 的反正切值。

```
SELECT ATAN( 0.52 );
```

## ATAN2 函数 [Numeric]

返回两个数字的比率的反正切值（以弧度表示）。

**语法**

**ATAN2** ( *numeric-expression-1*, *numeric-expression-2* )

**参数**

- **numeric-expression-1** 要计算其反正切值的比率中的分子。
- **numeric-expression-2** 要计算其反正切值的比率中的分母。

**返回值**

DOUBLE

**注释**

此函数将其参数转换为 DOUBLE，并以双精度浮点执行计算。

**另请参见**

- [“ACOS 函数 \[Numeric\]” 一节第 330 页](#)
- [“ASIN 函数 \[Numeric\]” 一节第 332 页](#)
- [“ATAN 函数 \[Numeric\]” 一节第 333 页](#)
- [“TAN 函数 \[Numeric\]” 一节第 409 页](#)

**标准和兼容性**

- **SQL/2003** 服务商扩充。

**示例**

以下语句返回 0.52 与 0.60 的比率的反正切值。

```
SELECT ATAN2( 0.52, 0.60 );
```



## AVG 函数 [Aggregate]

计算一组行、一个数字表达式或一组唯一值的平均值。

### 语法 1

```
AVG( numeric-expression | DISTINCT numeric-expression )
```

### 参数

- **numeric-expression** 要在一组行上计算其平均值的表达式。
- **DISTINCT 子句** 计算输入中的唯一数字值的平均值。

### 返回值

如果组不包含任何行，则返回 NULL 值。

如果参数为 DOUBLE，则返回 DOUBLE；否则返回 NUMERIC。

### 注释

此平均值不包括 *numeric-expression* 为 NULL 值的行。

### 另请参见

- [“SUM 函数 \[Aggregate\]” 一节第 406 页](#)
- [“COUNT 函数 \[Aggregate\]” 一节第 345 页](#)

### 标准和兼容性

- **SQL/2003** 核心特性。

### 示例

以下语句返回值 49988.623200。

```
SELECT AVG( Salary ) FROM Employees;
```

以下语句返回 Products 表中的平均产品价格：

```
SELECT AVG( DISTINCT UnitPrice ) FROM Products;
```

## BYTE\_LENGTH 函数 [String]

返回字符串中的字节数。

### 语法

```
BYTE_LENGTH( string-expression )
```

### 参数

- **string-expression** 要计算其长度的字符串。

## 返回值

INT

## 注释

返回的长度中包括 *string-expression* 中的尾随空格字符。

NULL 字符串的返回值为 NULL。

如果字符串使用的是多字节字符集，则 BYTE\_LENGTH 值可能不同于 CHAR\_LENGTH 返回的字符数。

## 另请参见

- “CHAR\_LENGTH 函数 [String]” 一节第 340 页
- “DATALENGTH 函数 [System]” 一节第 346 页
- “LENGTH 函数 [String]” 一节第 370 页
- “UltraLite 字符串函数” 一节第 327 页

## 标准和兼容性

- SQL/2003 服务商扩充。

## 示例

以下语句返回值 12。

```
SELECT BYTE_LENGTH( 'Test Message' );
```

# BYTE\_SUBSTR 函数 [String]

返回字符串的子串。子串是用字节而不是字符来计算的。

## 语法

```
BYTE_SUBSTR( string-expression, start [, length ] )
```

## 参数

- **string-expression** 从中获取子串的字符串。
- **start** 表示子串开始位置的整数表达式。正整数表示从字符串开始处开始，第一个字符位置为 1。负整数表示子串从字符串结尾处开始，最后一个字符的位置为 -1。
- **length** 表示子串长度的整数表达式。正 *length* 表示从开始位置向右的字节数。负 *length* 表示至多返回开始位置左侧（包括开始位置）的 *length* 个字节。

## 返回值

返回的值取决于 *string-expression* 的类型。此外，所指定的参数决定返回的值是否为 LONG。例如，当您为长度指定小于 32K 的常量时，不会返回 LONG。

BINARY

VARCHAR

## NVARCHAR

### 注释

如果指定了 *length*，则子串不能超过该字节数。*start* 和 *length* 可以是正数也可以是负数。通过使用适当的负数和正数组合，可以从字符串开始处或结束处获取子串。

如果 *start* 为零且 *length* 为非负数，则使用 1 作为 *start* 的值。如果 *start* 为零且 *length* 为负数，则使用 -1 作为 *start* 值。

### 另请参见

- [“SUBSTRING 函数 \[String\]”一节第 405 页](#)
- [“UltraLite 字符串函数”一节第 327 页](#)

### 标准和兼容性

- **SQL/2003** 服务商扩充。

### 示例

以下语句返回值 Test。

```
SELECT BYTE_SUBSTR( 'Test Message', 1, 4 );
```

## CAST 函数 [Data type conversion]

返回转换为提供的数据类型的表达式的值。

CAST、CONVERT、HEXTOINT、和 INTTOHEX 函数可用于十六进制值与其它值之间的相互转换。有关使用这些函数的详细信息，请参见“[转换至/自十六进制值](#)”一节《[SQL Anywhere 服务器 - SQL 参考](#)》。

### 语法

```
CAST( expression AS datatype )
```

### 参数

- **expression** 要转换的表达式。
- **data type** 目标数据类型。

### 返回值

指定的数据类型。

### 注释

如果不指示字符串类型的长度，数据库服务器会选择适当的长度。如果没有为 DECIMAL 转换指定精度和小数位数，数据库服务器会选择适当的值。

### 另请参见

- [“CONVERT 函数 \[Data type conversion\]”](#) 一节第 342 页
- [“LEFT 函数 \[String\]”](#) 一节第 369 页

### 标准和兼容性

- **SQL/2003** 核心特性。

### 示例

以下函数确保字符串被用作日期：

```
SELECT CAST( '2000-10-31' AS DATE );
```

计算表达式  $1 + 2$  的值，并将结果转换为单字符字符串。

```
SELECT CAST( 1 + 2 AS CHAR );
```

可使用 CAST 函数缩短字符串长度

```
SELECT CAST ( 'Surname' AS CHAR(5) );
```

## CEILING 函数 [Numeric]

返回第一个大于或等于给定值的整数。对于正数，这被称为向上舍入。

### 语法

```
CEILING( numeric-expression )
```

### 参数

- **numeric-expression** 要计算其上限的数字。

### 返回值

DOUBLE

### 注释

此函数将其参数转换为 DOUBLE，并以双精度浮点执行计算。

### 另请参见

- [“FLOOR 函数 \[Numeric\]”](#) 一节第 360 页

### 标准和兼容性

- **SQL/2003** 服务商扩充。

### 示例

以下语句返回值 60。

```
SELECT CEILING( 59.84567 );
```

## CHAR 函数 [String]

返回数字所表示的 ASCII 码值的字符。

### 语法

**CHAR**( *integer-expression* )

### 参数

- **integer-expression** 要转换成 ASCII 字符的数字。该数字必须在 0 到 255 范围内（含 0 和 255）。

### 返回值

VARCHAR

### 注释

根据二进制排序顺序，返回的字符与当前数据库的字符集中提供的数字表达式相对应。

对于值大于 255 或小于 0 的整数表达式，CHAR 返回 NULL。

### 另请参见

- [“UltraLite 字符串函数”一节第 327 页](#)

### 标准和兼容性

- **SQL/2003** 服务商扩充。

### 示例

以下语句返回值 Y。

```
SELECT CHAR( 89 );
```

## CHARINDEX 函数 [String]

返回一个字符串在另一个字符串中的位置。

### 语法

**CHARINDEX**( *string-expression-1*, *string-expression-2* )

### 参数

- **string-expression-1** 要搜索的字符串。
- **string-expression-2** 被搜索的字符串。

### 返回值

INT

## 注释

*string-expression-1* 的第一个字符标识为 1。如果被搜索的字符串包含一个以上另一字符串的实例，则 CHARINDEX 函数返回第一个实例的位置。

如果被搜索的字符串不包含另一个字符串，则 CHARINDEX 函数返回 0。

## 另请参见

- “SUBSTRING 函数 [String]” 一节第 405 页
- “REPLACE 函数 [String]” 一节第 391 页
- “LOCATE 函数 [String]” 一节第 372 页
- “UltraLite 字符串函数” 一节第 327 页

## 标准和兼容性

- **SQL/2003** 服务商扩充。

## 示例

以下语句仅在姓氏中包含字母 K 时才会从 Employees 表中的 Surname 和 GivenName 列返回姓和名：

```
SELECT Surname, GivenName
FROM Employees
WHERE CHARINDEX( 'K', Surname ) = 1;
```

返回的结果：

Surname	GivenName
Klobucher	James
Kuo	Felicia
Kelly	Moira

## CHAR\_LENGTH 函数 [String]

返回字符串中的字符数。

## 语法

**CHAR\_LENGTH** ( *string-expression* )

## 参数

- **string-expression** 要计算其长度的字符串。

## 返回值

INT

## 注释

返回的长度中包括尾随空格字符。

NULL 字符串的返回值为 NULL。

如果字符串使用的是多字节字符集，则 CHAR\_LENGTH 函数返回的值可能不同于 BYTE\_LENGTH 函数返回的字节数。

### 注意

对于 CHAR、VARCHAR 和 LONG VARCHAR 数据类型，可以交替使用 CHAR\_LENGTH 函数和 LENGTH 函数。不过，对于 BINARY 和位数组数据类型，必须使用 LENGTH 函数。

## 另请参见

- “[BYTE\\_LENGTH 函数 \[String\]](#)” 一节第 335 页
- “[UltraLite 字符串函数](#)” 一节第 327 页

## 标准和兼容性

- **SQL/2003** 核心特性。

## 示例

以下语句返回值 8。

```
SELECT CHAR_LENGTH( 'Chemical' );
```

# COALESCE 函数 [Miscellaneous]

返回列表中的第一个非 NULL 表达式。此函数与 ISNULL 函数完全相同。

## 语法

```
COALESCE( expression, expression [ , ... ] )
```

## 参数

- **expression** 任意表达式。

必须至少给此函数传递两个表达式，并且所有表达式必须可以进行比较。

## 返回值

ANY

## 注释

仅当所有参数都为 NULL 时结果才是 NULL。

参数可以是任何标量类型，但不必是同一类型。

有关数据库服务器对此函数处理方法的详细说明，请参见“[ISNULL 函数 \[Miscellaneous\]](#)”一节第 367 页。

**另请参见**

- “ISNULL 函数 [Miscellaneous]” 一节第 367 页

**标准和兼容性**

- **SQL/2003** 核心特性。

**示例**

以下语句返回值 34。

```
SELECT COALESCE( NULL, 34, 13, 0 );
```

## CONVERT 函数 [Data type conversion]

返回转换成提供的数据类型的表达式。

CAST、CONVERT、HEXTTOINT、和 INTTOHEX 函数可用于十六进制值与其它值之间的相互转换。有关使用这些函数的详细信息，请参见“[转换至/自十六进制值](#)”一节《[SQL Anywhere 服务器 - SQL 参考](#)》。

**语法**

```
CONVERT( datatype, expression [ , format-style ] )
```

**参数**

- **datatype** 表达式将要转换成的数据类型。
- **expression** 要转换的表达式。
- **format-style** 要应用于输出值的样式代码。请在将字符串转换为日期或时间数据类型或将日期或时间数据类型转换为字符串时使用此参数。下表显示了所支持的样式代码，其后为该样式代码产生的输出格式表示。样式代码分为两列，具体要视输出格式中是否包括世纪而定（例如，不包括世纪时为 06，包括世纪时则为 2006）。

不含世纪 (yy) 的样式代码	含世纪 (yyyy) 的样式代码	输出格式
-	0 或 100	Mmm dd yyyy hh:nnAA
1	101	mm/dd/yy[yy]
2	102	[yy]yy.mm.dd
3	103	dd/mm/yy[yy]
4	104	dd.mm.yy[yy]
5	105	dd-mm-yy[yy]



不含世纪 (yy) 的样式代码	含世纪 (yyyy) 的样式代码	输出格式
6	106	dd Mmm yy[yy]
7	107	Mmm dd, yy[yy]
8	108	hh:nn:ss
-	9 或 109	Mmm dd yyyy hh:nn:ss:sssAA
10	110	mm-dd-yy[yy]
11	111	[yy]yy/mm/dd
12	112	[yy]yyymmdd
-	13 或 113	dd Mmm yyyy hh:nn:ss:sss (24 小时制, 欧洲缺省时间 + 毫秒, 4 位数年份)
-	14 或 114	hh:nn:ss:sss (24 小时制)
-	20 或 120	yyyy-mm-dd hh:nn:ss (24 小时制, ODBC 规范, 4 位数年份)
-	21 或 121	yyyy-mm-dd hh:nn:ss:sss (24 小时制, ODBC 规范加毫秒, 4 位数年份)

### 返回值

指定的数据类型。

### 注释

如果不提供 *format-style* 参数, 将使用样式代码 0。

有关每种输出符号 (如 Mmm) 生成的样式的说明, 请参见 [“UltraLite date\\_format 创建参数”](#) 一节第 171 页。

### 另请参见

- [“CAST 函数 \[Data type conversion\]”](#) 一节第 337 页

### 标准和兼容性

- **SQL/2003** 服务商扩充。

### 示例

以下语句举例说明了格式样式的使用法。

```
SELECT CONVERT( CHAR( 20 ), OrderDate, 104 ) FROM SalesOrders;
```

OrderDate
16.03.2000
20.03.2000
23.03.2000
25.03.2000
...

```
SELECT CONVERT( CHAR( 20 ), OrderDate, 7 ) FROM SalesOrders;
```

OrderDate
Mar 16, 00
Mar 20, 00
Mar 23, 00
Mar 25, 00
...

以下语句举例说明了到整数的转换，并返回值 5。

```
SELECT CONVERT( integer, 5.2 );
```

## COS 函数 [Numeric]

返回由其参数所指定的角度（以弧度表示）的余弦值。

### 语法

```
COS( numeric-expression )
```

### 参数

- **numeric-expression** 角度（以弧度为单位）。

### 返回值

此函数将其参数转换为 DOUBLE，以双精度浮点执行计算，然后返回 DOUBLE 值作为结果。如果参数为 NULL，则结果为 NULL。

### 另请参见

- “ACOS 函数 [Numeric]” 一节第 330 页
- “COT 函数 [Numeric]” 一节第 345 页
- “SIN 函数 [Numeric]” 一节第 399 页
- “TAN 函数 [Numeric]” 一节第 409 页

### 标准和兼容性

- **SQL/2003** 服务商扩充。

### 示例

以下语句返回 0.52 弧度角度的余弦值。

```
SELECT COS ( 0.52 );
```

## COT 函数 [Numeric]

返回由其参数所指定的角度（以弧度表示）的余切值。

### 语法

**COT**( *numeric-expression* )

### 参数

- **numeric-expression** 角度（以弧度为单位）。

### 返回值

此函数将其参数转换为 DOUBLE，以双精度浮点执行计算，然后返回 DOUBLE 值作为结果。如果参数为 NULL，则结果为 NULL。

### 另请参见

- “COS 函数 [Numeric]” 一节第 344 页
- “SIN 函数 [Numeric]” 一节第 399 页
- “TAN 函数 [Numeric]” 一节第 409 页

### 标准和兼容性

- **SQL/2003** 服务商扩充。

### 示例

以下语句返回 0.52 的余切值。

```
SELECT COT ( 0.52 );
```

## COUNT 函数 [Aggregate]

根据指定的参数计算组中的行数。

**语法 1**

```
COUNT(  
*  
| expression  
| DISTINCT expression  
)
```

**参数**

- \* 返回每组中的行数。
- **expression** 要返回行数的表达式。
- **DISTINCT expression** 要返回非重复行数的表达式。

**返回值**

INT

**注释**

计数中不包括值为 NULL 的行。

**另请参见**

- [“AVG 函数 \[Aggregate\]” 一节第 335 页](#)
- [“SUM 函数 \[Aggregate\]” 一节第 406 页](#)

**标准和兼容性**

- **SQL/2003** 核心特性。语法 2 为特性 T611。

**示例**

以下语句返回每个唯一的城市和含有该城市值的行数。

```
SELECT City, COUNT( * ) FROM Employees GROUP BY City;
```

## DATALENGTH 函数 [System]

返回表达式结果的基础存储的长度（以字节为单位）。

**语法**

```
DATALENGTH( expression )
```

**参数**

- **expression** 通常是列的名称。如果 *expression* 是字符串常量，则必须将它用引号括起来。

**返回值**

UNSIGNED INT

**注释**

DATALENGTH 函数的返回值如下：

数据类型	DATALENGTH
SMALLINT	2
INTEGER	4
DOUBLE	8
CHAR	数据的长度
BINARY	数据的长度

**标准和兼容性**

- **SQL/2003** 服务商扩充。

**示例**

以下语句返回 CompanyName 列中最长字符串的长度。

```
SELECT MAX( DATALENGTH( CompanyName ) )
FROM Customers;
```

以下语句返回字符串 '8sdofinsv8s7a7s7gehe4h' 的长度：

```
SELECT DATALENGTH( '8sdofinsv8s7a7s7gehe4h' );
```

## DATE 函数 [Date and time]

将表达式转换为日期，并删除任何小时、分钟或秒。

有关控制日期格式的解释的信息，请参见“[UltraLite date\\_order 创建参数](#)”一节第 174 页。

**语法**

**DATE**( *expression* )

**返回值**

DATE

**参数**

- **expression** 要转换为日期格式的值，通常为字符串。

**标准和兼容性**

- **SQL/2003** 服务商扩充。

## 示例

以下语句返回值 1999-01-02 作为日期。

```
SELECT DATE( '1999-01-02 21:20:53' );
```

以下语句返回 SYSOBJECT 系统视图中列出的所有对象的创建日期:

```
SELECT DATE( creation_time ) FROM SYSOBJECT;
```

## DATEADD 函数 [Date and time]

返回通过将多个日期部分添加到日期中而产生的日期。

### 语法

```
DATEADD( date-part, numeric-expression, date-expression )
```

*date-part* :

```
year  
| quarter  
| month  
| week  
| day  
| dayofyear  
| hour  
| minute  
| second  
| millisecond
```

### 参数

- **date-part** 要添加到日期中的日期部分。有关日期部分的详细信息，请参见“日期部分”一节第 325 页。
- **numeric-expression** 要添加到日期中的日期部分的数量。*numeric-expression* 可以是任何数字类型，但其值将截断为整数。
- **date-expression** 要修改的日期。

### 返回值

TIMESTAMP

### 标准和兼容性

- **SQL/2003** 服务商扩充。

### 示例

以下语句返回值: 1995-11-02 00:00:00.000.

```
SELECT DATEADD( month, 102, '1987/05/02' );
```

## DATEDIFF 函数 [Date and time]

返回两个日期之间的间隔。

### 语法

**DATEDIFF**( *date-part*, *date-expression-1*, *date-expression-2* )

*date-part* :  
year  
| quarter  
| month  
| week  
| day  
| dayofyear  
| hour  
| minute  
| second  
| millisecond

### 参数

- **date-part** 指定要测量间隔的日期部分。  
选择上面列出的日期对象之一。有关日期部分的完整列表，请参见“日期部分”一节第 325 页。
- **date-expression-1** 间隔的起始日期。从 *date-expression-2* 中减去此值，以返回这两个参数之间 *date-part* 的数量。
- **date-expression-2** 间隔的结束日期。从此值中减去 *date-expression-1*，以返回这两个参数之间 *date-part* 的数量。

### 返回值

INT

### 注释

此函数计算两个指定日期之间日期部分的数量。结果是等于 *date2* 与 *date1* 之差的有符号整数值（以日期部分为单位）。

当结果不是日期部分的偶数倍时，将会截断而不是舍入 DATEDIFF 函数的结果。

当使用 **day** 作为日期部分时，DATEDIFF 函数返回两个指定时间之间（包括第二个日期但不包括第一个日期）的午夜数。

当使用 **month** 作为日期部分时，DATEDIFF 函数返回两个日期之间（包括第二个日期但不包括第一个日期）出现的月第一天的个数。

当使用 **week** 作为日期部分时，DATEDIFF 函数返回两个日期之间（包括第二个日期但不包括第一个日期）的星期日的个数。

对于更小的时间单位存在溢出值：

- **millisecond** 24 天
- **seconds** 68 年

- **minute** 4083 年
- **others** 没有溢出限制

如果超出这些限制，此函数将返回溢出错误。

### 标准和兼容性

- **SQL/2003** Transact-SQL 扩充。

### 示例

以下语句返回 1。

```
SELECT DATEDIFF( hour, '4:00AM', '5:50AM' );
```

以下语句返回 102。

```
SELECT DATEDIFF( month, '1987/05/02', '1995/11/15' );
```

以下语句返回 0。

```
SELECT DATEDIFF( day, '00:00', '23:59' );
```

以下语句返回 4。

```
SELECT DATEDIFF( day,  
                '1999/07/19 00:00',  
                '1999/07/23 23:59' );
```

以下语句返回 0。

```
SELECT DATEDIFF( month, '1999/07/19', '1999/07/23' );
```

以下语句返回 1。

```
SELECT DATEDIFF( month, '1999/07/19', '1999/08/23' );
```

## DATEFORMAT 函数 [Date and time]

返回以指定格式表示日期表达式的字符串。

### 语法

```
DATEFORMAT( datetime-expression, string-expression )
```

### 参数

- **datetime-expression** 要转换的日期时间。
- **string-expression** 转换后的日期格式。

有关日期格式说明的信息，请参见“[UltraLite date\\_format 创建参数](#)”一节第 171 页。

### 返回值

VARCHAR



## 注释

任何允许的日期格式都可用于字符串表达式。

## 标准和兼容性

- **SQL/2003** 服务商扩充。

## 示例

以下语句返回值 Jan 01, 1989。

```
SELECT DATEFORMAT( '1989-01-01', 'Mmm dd, yyyy' );
```

## DATENAME 函数 [Date and time]

以字符串形式返回日期时间值中指定部分的名称（如月份 June）。

## 语法

```
DATENAME( date-part, date-expression )
```

## 参数

- **date-part** 要指定的日期部分。  
有关允许的日期部分的完整列表，请参见“日期部分”一节第 325 页。
- **date-expression** 要返回日期部分名的日期。该日期必须包含所请求的 *date-part*。

## 返回值

VARCHAR

## 注释

DATENAME 函数返回字符串，即使结果是数字类型（如表示日期的 23）。

## 标准和兼容性

- **SQL/2003** 服务商扩充。

## 示例

以下语句返回值 May。

```
SELECT DATENAME( month, '1987/05/02' );
```

## DATEPART 函数 [Date and time]

返回日期时间值中一部分的值。

## 语法

```
DATEPART( date-part, date-expression )
```

**参数**

- **date-part** 要返回的日期部分。  
有关允许的日期部分的完整列表，请参见“日期部分”一节第 325 页。
- **date-expression** 要返回的日期部分的日期。

**返回值**

INT

**注释**

该日期必须包含 *date-part* 字段。

**标准和兼容性**

- **SQL/2003** 服务商扩充。

**示例**

以下语句返回值 5。

```
SELECT DATEPART( month , '1987/05/02' );
```

下面的示例将创建表 TableStatistics，并向其中插入 SalesOrders 表中所存储的每年销售订单总数：

```
CREATE TABLE TableStatistics (
    ID INTEGER NOT NULL DEFAULT AUTOINCREMENT,
    Year INT,
    NumberOrders INT );
INSERT INTO TableStatistics ( Year, NumberOrders )
SELECT DATEPART( Year, OrderDate ), COUNT(*)
FROM SalesOrders
GROUP BY DATEPART( Year, OrderDate );
```

## DATETIME 函数 [Date and time]

将表达式转换成时间戳。

**语法**

```
DATETIME( expression )
```

**参数**

- **expression** 要转换的表达式。通常是字符串。

**返回值**

TIMESTAMP

**注释**

试图转换数字值时会返回错误。

### 标准和兼容性

- **SQL/2003** 服务商扩充。

### 示例

以下语句返回值为 1998-09-09 12:12:12.000 的时间戳。

```
SELECT DATETIME( '1998-09-09 12:12:12.000' );
```

## DAY 函数 [Date and time]

返回 1 到 31 之间的一个整数。

### 语法

```
DAY( date-expression )
```

### 参数

- **date-expression** 日期。

### 返回值

SMALLINT

### 注释

整数 1 到 31 对应于日期中的月号。

### 标准和兼容性

- **SQL/2003** 服务商扩充。

### 示例

以下语句返回值 12。

```
SELECT DAY( '2001-09-12' );
```

## DAYNAME 函数 [Date and time]

返回日期中的星期几的名称。

### 语法

```
DAYNAME( date-expression )
```

### 参数

- **date-expression** 日期。

### 返回值

VARCHAR

## 注释

返回的英语名如下：Sunday、Monday、Tuesday、Wednesday、Thursday、Friday、Saturday。

## 标准和兼容性

- **SQL/2003** 服务商扩充。

## 示例

以下语句返回值 Saturday。

```
SELECT DAYNAME ( '1987/05/02' );
```

# DAYS 函数 [Date and time]

计算天数的函数。有关具体详细信息，请参见此函数的用法。

## 语法 1: integer

```
DAYS( [ datetime-expression, ] datetime-expression )
```

## 语法 2: timestamp

```
DAYS( datetime-expression, integer-expression )
```

## 参数

- **datetime-expression** 日期和时间。
- **integer-expression** 要添加到 *datetime-expression* 中的天数。如果 *integer-expression* 是负数，则从时间戳中减去相应的天数。如果提供整数表达式，则必须将 *datetime-expression* 显式地转换为日期或时间戳。

有关转换数据类型的信息，请参见“CAST 函数 [Data type conversion]”一节第 337 页。

## 返回值

INT，如果指定两个日期时间表达式。

TIMESTAMP，如果指定的第二个参数为整数。

## 注释

此函数的行为会视所指定信息的不同而变化：

- 如果提供一个日期，此函数返回自 0000-02-29 以来的天数。

### 注意

0000-02-29 并不表示实际日期，它是日期算法使用的日期。

- 如果提供两个日期，此函数返回它们之间的整数天数。替代方法为使用 DATEDIFF 函数。
- 如果提供一个日期和一个整数，此函数会为指定日期添加整数天数。替代方法为使用 DATEADD 函数。

此函数会忽略小时、分钟和秒。

### 另请参见

- “DATEDIFF 函数 [Date and time]” 一节第 349 页
- “DATEADD 函数 [Date and time]” 一节第 348 页

### 标准和兼容性

- **SQL/2003** 服务商扩充。

### 示例

以下语句返回整数 729889。

```
SELECT DAYS( '1998-07-13 06:07:12' );
```

以下语句返回整数值 -366，表示第二个日期比第一个日期早 366 天。建议使用第二个示例 (DATEDIFF)。

```
SELECT DAYS( '1998-07-13 06:07:12',  
            '1997-07-12 10:07:12' );
```

```
SELECT DATEDIFF( day,  
                '1998-07-13 06:07:12',  
                '1997-07-12 10:07:12' );
```

以下语句返回时间戳 14.07.99 00:00:000.000。建议使用第二个示例 (DATEADD)。

```
SELECT DAYS( CAST('1998-07-13' AS DATE ), 366 );  
SELECT DATEADD( day, 366, '1998-07-13' );
```

## DB\_PROPERTY 函数 [System]

返回给定属性的值。

### 语法

```
DB_PROPERTY( property-name )
```

### 参数

- **property-name** 数据库属性名称。

### 返回值

VARCHAR

### 注释

返回字符串。

要在 UltraLite 中设置选项，请使用 SET OPTION 语句或组件的特定于 API 的设置数据库选项方法。

### 另请参见

- [“UltraLite SET OPTION 语句”一节第 459 页](#)
- UltraLite C++: [“SetDatabaseOption 函数”一节](#) 《UltraLite - C 及 C++ 编程》
- UltraLite.NET: [“SetDatabaseOption 方法”一节](#) 《UltraLite - .NET 编程》

### 示例

以下语句返回当前数据库的页大小（以字节为单位）。

```
SELECT DB_PROPERTY( 'page_size');
```

## DEGREES 函数 [Numeric]

将数字从弧度转换为度数。

### 语法

```
DEGREES( numeric-expression )
```

### 参数

- **numeric-expression** 以弧度表示的角度。

### 返回值

返回由 *numeric-expression* 所指定的角的度数。

DOUBLE

### 注释

此函数将其参数转换为 DOUBLE，并以双精度浮点执行计算。如果参数为 NULL，则结果为 NULL。

### 标准和兼容性

- **SQL/2003** 服务商扩充。

### 示例

以下语句返回值 29.79380534680281。

```
SELECT DEGREES( 0.52 );
```

## DIFFERENCE 函数 [String]

返回两个字符串表达式之间 SOUNDEX 值的差。

### 语法

```
DIFFERENCE ( string-expression-1, string-expression-2 )
```

### 参数

- **string-expression-1** 第一个 SOUNDEX 参数。
- **string-expression-2** 第二个 SOUNDEX 参数。

### 返回值

SMALLINT

### 注释

DIFFERENCE 函数对两个字符串的 SOUNDEX 值进行比较，并计算它们之间的相似程度，返回 0 到 4 之间的值，4 表示最佳匹配。

此函数总是会返回某个值。仅当其中一个参数是 NULL 时，结果才是 NULL。

### 另请参见

- [“SOUNDEX 函数 \[String\]”一节第 400 页](#)
- [“UltraLite 字符串函数”一节第 327 页](#)

### 标准和兼容性

- **SQL/2003** 服务商扩充。

### 示例

以下语句返回值 3。

```
SELECT DIFFERENCE( 'test', 'chest' );
```

## DOW 函数 [Date and time]

返回 1 到 7 之间的一个数字，表示某个日期是星期几，其中 1 表示星期日，2 表示星期一，依此类推。

### 语法

**DOW**( *date-expression* )

### 参数

- **date-expression** 要计算的日期。

### 返回值

SMALLINT

### 注释

DOW 函数不受为 `first_day_of_week` 数据库选项指定的值的影响。例如，即使将 `first_day_of_week` 设置为 Monday，对于 Monday，DOW 函数仍会返回 2。

### 标准和兼容性

- **SQL/2003** 服务商扩充。

### 示例

以下语句返回值 5。

```
SELECT DOW( '1998-07-09' );
```

以下语句查询 Employees 表，然后返回雇员的 StartDate，以与星期几对应的数字表示：

```
SELECT DOW( StartDate ) FROM Employees;
```



## SQL 函数 (E-O)

列出了每个函数，并在它旁边指出函数类型（数字、字符等等）。

有关指向给定类型所有函数的链接，请参见“函数类型”一节第 324 页。

### 另请参见

- “SQL 函数 (A-D)”一节 《SQL Anywhere 服务器 - SQL 参考》
- “SQL 函数 (P-Z)”一节 《SQL Anywhere 服务器 - SQL 参考》

## EXP 函数 [Numeric]

返回指数函数，即 e 的指定数字次乘方。

### 语法

**EXP**( *numeric-expression* )

### 参数

- **numeric-expression** 指数。

### 返回值

DOUBLE

### 注释

EXP 函数返回由 *numeric-expression* 指定的值的指数。

此函数将其参数转换为 DOUBLE，以双精度浮点执行计算，然后返回 DOUBLE 值作为结果。如果参数为 NULL，则结果为 NULL。

### 标准和兼容性

- **SQL/2003** 服务商扩充。

### 示例

以下语句返回值 3269017.3724721107。

```
SELECT EXP( 15 );
```

## EXPLANATION 函数 [Miscellaneous]

返回 SQL 语句的计划优化策略。

### 语法

**EXPLANATION**( *string-expression* )

**参数**

- **string-expression** SQL 语句，通常是 SELECT 语句，但也可以是 UPDATE 或 DELETE 语句。

**返回值**

LONG VARCHAR

**注释**

优化以字符串形式返回。

这些信息可以帮助决定要添加哪些索引或如何构建数据库，以获得更好的性能。

**另请参见**

- [“UltraLite 中的执行计划”一节第 320 页](#)

**标准和兼容性**

- **SQL/2003** 服务商扩充。

**示例**

下面的语句将 SELECT 语句作为字符串参数传递，并返回查询的执行计划。

```
SELECT EXPLANATION( 'SELECT * FROM Departments WHERE DepartmentID > 100' );
```

## FLOOR 函数 [Numeric]

返回不大于某个数字的最大整数。

**语法**

```
FLOOR( numeric-expression )
```

**参数**

- **numeric-expression** 值，通常为 FLOAT。

**返回值**

DOUBLE

**注释**

此函数将其参数转换为 DOUBLE，并以双精度浮点执行计算。

**另请参见**

- [“CEILING 函数 \[Numeric\]”一节第 338 页](#)

**标准和兼容性**

- **SQL/2003** 服务商扩充。

## 示例

以下语句返回 Floor 值 123。

```
SELECT FLOOR (123);
```

以下语句返回 Floor 值 123。

```
SELECT FLOOR (123.45);
```

以下语句返回 Floor 值 -124。

```
SELECT FLOOR (-123.45);
```

## GETDATE 函数 [Date and time]

返回当前的年、月、日、小时、分钟、秒和秒的小数部分。

### 语法

```
GETDATE()
```

### 返回值

TIMESTAMP

### 注释

其准确性受系统时钟准确性的限制。

GETDATE 函数返回的信息与 NOW 函数返回的信息及 CURRENT\_TIMESTAMP 特殊值相同。

### 另请参见

- [“NOW 函数 \[Date and time\]” 一节第 384 页](#)

### 标准和兼容性

- **SQL/2003** 服务商扩充。

### 示例

以下语句返回系统日期和时间。

```
SELECT GETDATE ( );
```

## GREATER 函数 [Miscellaneous]

返回两个参数值中的较大参数值。

### 语法

```
GREATER( expression-1, expression-2 )
```

**参数**

- **expression-1** 要比较的第一个参数值。
- **expression-2** 要比较的第二个参数值。

**返回值**

ANY

**注释**

如果两个参数相等，则返回第一个。

**另请参见**

- [“LESSER 函数 \[Miscellaneous\]” 一节第 370 页](#)

**标准和兼容性**

- **SQL/2003** 服务商扩充。

**示例**

以下语句返回值 10。

```
SELECT GREATER( 10, 5 ) FROM dummy;
```

## HEXTOINT 函数 [Data type conversion]

返回十六进制字符串的等效十进制整数。

CAST、CONVERT、HEXTOINT、和 INTTOHEX 函数可用于十六进制值与其它值之间的相互转换。有关使用这些函数的详细信息，请参见“[转换至/自十六进制值](#)”一节《[SQL Anywhere 服务器 - SQL 参考](#)》。

**语法**

```
HEXTOINT( hexadecimal-string )
```

**参数**

- **hexadecimal-string** 要转换成整数的字符串。

**返回值**

HEXTOINT 函数返回十六进制字符串的等效平台无关 SQL INTEGER。如果从右侧数第 8 位数是数字 8-9 以及大写或小写字母 A-F 中的某一个，并且前面的前导位都是大写或小写字母 F，则该十六进制值代表负整数。以下语句中对 HEXTOINT 的使用无效，因为该参数表示的正整数值无法表示为有符号 32 位整数：

```
SELECT HEXTOINT( '0x0080000001' );
```

INT

## 注释

HEXTOINT 函数只接受由数字以及大写或小写字母 A-F 组成的字符串文字或字符串变量（带有或不带有 0x 前缀）。以下都是有效的 HEXTOINT 用法：

```
SELECT HEXTOINT( '0xFFFFFFFF' );
SELECT HEXTOINT( '0x00000100' );
SELECT HEXTOINT( '100' );
SELECT HEXTOINT( '0xfffffffff80000001' );
```

存在 0x 前缀时，HEXTOINT 函数会将其删除。如果数据超过 8 位数，则它所表示的值必须可以表示为有符号 32 位整数值。

## 另请参见

- “[INTTOHEX 函数 \[Data type conversion\]](#)” 一节第 366 页

## 标准和兼容性

- **SQL/2003** 服务商扩充。

## 示例

以下语句返回值 420。

```
SELECT HEXTOINT( '1A4' );
```

# HOUR 函数 [Date and time]

返回日期时间的小时部分。

## 语法

**HOUR**( *datetime-expression* )

## 参数

- **datetime-expression** 日期时间。

## 返回值

SMALLINT

## 注释

返回的值是 0 到 23 之间的某个数字，对应于日期时间的小时。

## 标准和兼容性

- **SQL/2003** 服务商扩充。

## 示例

以下语句返回值 21：

```
SELECT HOUR( '1998-07-09 21:12:13' );
```

## HOURS 函数 [Date and time]

计算小时的函数。有关具体详细信息，请参见此函数的用法。

### 语法 1: integer

**HOURS** ( [ *datetime-expression*, ] *datetime-expression* )

### 语法 2: timestamp

**HOURS** ( *datetime-expression*, *integer-expression* )

### 参数

- **datetime-expression** 日期和时间。
- **integer-expression** 要添加到 *datetime-expression* 中的小时数。如果 *integer-expression* 是负数，则从日期时间中减去相应的小时数。如果提供整数表达式，则必须将 *datetime-expression* 显式地转换为 DATETIME 数据类型。

有关转换数据类型的信息，请参见“CAST 函数 [Data type conversion]”一节第 337 页。

### 返回值

INT

TIMESTAMP

### 注释

所指定的信息会更改此函数的行为：

- 如果提供一个日期，此函数返回自 0000-02-29 以来的小时数。

#### 注意

0000-02-29 并不表示实际日期，它是日期算法使用的日期。

- 如果提供两个时间戳，此函数会返回它们之间的整数小时数。替代方法为使用 DATEDIFF 函数。
- 如果提供一个日期和一个整数，此函数会为指定时间戳添加整数小时数。替代方法为使用 DATEADD 函数。

### 另请参见

- “DATEDIFF 函数 [Date and time]”一节第 349 页
- “DATEADD 函数 [Date and time]”一节第 348 页

### 标准和兼容性

- **SQL/2003** 服务商扩充。

## 示例

以下语句返回值 4，表示第二个时间戳比第一个时间戳晚四个小时。建议使用第二个示例 (DATEDIFF)。

```
SELECT HOURS( '1999-07-13 06:07:12',
              '1999-07-13 10:07:12' );

SELECT DATEDIFF( hour,
                 '1999-07-13 06:07:12',
                 '1999-07-13 10:07:12' );
```

以下语句返回值 17517342。

```
SELECT HOURS( '1998-07-13 06:07:12' );
```

以下语句返回日期时间 13.05.99 02:05:070.000。建议使用第二个示例 (DATEADD)。

```
SELECT HOURS (
    CAST( '1999-05-12 21:05:07' AS DATETIME ), 5 );

SELECT DATEADD( hour, 5, '1999-05-12 21:05:07' );
```

## IFNULL 函数 [Miscellaneous]

如果第一个表达式为 NULL 值，则返回第二个表达式的值。如果第一个表达式不为 NULL，则返回第三个表达式的值。如果第一个表达式不为 NULL，并且没有第三个表达式，则返回 NULL。

### 语法

```
IFNULL( expression-1, expression-2 [ , expression-3 ] )
```

### 参数

- **expression-1** 要计算的表达式。它的值决定返回 *expression-2* 还是 *expression-3*。
- **expression-2** *expression-1* 为 NULL 时的返回值。
- **expression-3** *expression-1* 不为 NULL 时的返回值。

### 返回值

返回的数据类型取决于 *expression-2* 和 *expression-3* 的数据类型。

### 标准和兼容性

- **SQL/2003** Transact-SQL 扩充。

### 示例

以下语句返回值 -66。

```
SELECT IFNULL( NULL, -66 );
```

以下语句返回 NULL，因为第一个表达式不为 NULL，并且没有第三个表达式。

```
SELECT IFNULL( -66, -66 );
```

## INSERTSTR 函数 [String]

将一个字符串插入到另一个字符串中的指定位置。

### 语法

```
INSERTSTR( integer-expression, string-expression-1, string-expression-2 )
```

### 参数

- **integer-expression** 要在其后插入字符串的位置。使用零可在起始处插入字符串。
- **string-expression-1** 要在其中插入其它字符串的字符串。
- **string-expression-2** 要插入的字符串。

### 返回值

LONG VARCHAR

### 另请参见

- [“STUFF 函数 \[String\]”一节第 404 页](#)
- [“UltraLite 字符串函数”一节第 327 页](#)

### 标准和兼容性

- **SQL/2003** 服务商扩充。

### 示例

以下语句返回值 backoffice。

```
SELECT INSERTSTR( 0, 'office ', 'back' );
```

## INTTOHEX 函数 [Data type conversion]

返回包含整数的等效十六进制值的字符串。

### 语法

```
INTTOHEX( integer-expression )
```

### 参数

- **integer-expression** 要转换成十六进制的整数。

### 返回值

VARCHAR

### 注释

CAST、CONVERT、HEXTOINT、和 INTTOHEX 函数可用于十六进制值与其它值之间的相互转换。有关详细信息，请参见 [“转换至/自十六进制值”一节](#) 《SQL Anywhere 服务器 - SQL 参考》。



### 另请参见

- “HEXTOINT 函数 [Data type conversion]” 一节第 362 页

### 标准和兼容性

- **SQL/2003** Transact-SQL 扩充。

### 示例

以下语句返回值 0000009c。

```
SELECT INTTOHEX( 156 );
```

## ISDATE 函数 [Data type conversion]

测试字符串参数是否可以转换为日期。

### 语法

```
ISDATE( string )
```

### 参数

- **string** 要分析的字符串（该字符串将被分析以确定是否代表了一个有效的日期）。

### 返回值

INT

### 注释

如果可以进行转换，则此函数返回 1；否则返回 0。如果参数为 NULL，则返回 0。

### 标准和兼容性

- **SQL/2003** 服务商扩充。

## ISNULL 函数 [Miscellaneous]

返回列表中的第一个非 NULL 表达式。此函数与 COALESCE 函数完全相同。

### 语法

```
ISNULL( expression, expression [, ...] )
```

### 参数

- **expression** 要测试是否为 NULL 的表达式。  
必须至少给此函数传递两个表达式，并且所有表达式必须可以进行比较。

## 返回值

此函数的返回类型取决于指定的表达式。即，数据库服务器计算此函数时，首先搜索可用于比较所有表达式的数据类型。如果找到通用类型，则数据库服务器比较表达式，然后以用于比较的类型返回结果。如果数据库服务器找不到通用比较类型，则返回错误。

## 另请参见

- [“COALESCE 函数 \[Miscellaneous\]” 一节第 341 页](#)

## 标准和兼容性

- **SQL/2003** 服务商扩充。

## 示例

以下语句返回值 -66。

```
SELECT ISNULL( NULL , -66, 55, 45, NULL, 16 );
```

# LCASE 函数 [String]

将字符串中的所有字符转换成小写形式。此函数与 LOWER 函数完全相同。

## 语法

**LCASE**( *string-expression* )

## 参数

- **string-expression** 要转换成小写形式的字符串。

## 返回值

CHAR

NCHAR

LONG VARCHAR

VARCHAR

NVARCHAR

## 注释

LCASE 函数类似于 LOWER 函数。

## 另请参见

- [“LOWER 函数 \[String\]” 一节第 374 页](#)
- [“UCASE 函数 \[String\]” 一节第 413 页](#)
- [“UPPER 函数 \[String\]” 一节第 414 页](#)
- [“UltraLite 字符串函数” 一节第 327 页](#)

## 标准和兼容性

- **SQL/2003** 服务商扩充。

## 示例

以下语句返回值 chocolate。

```
SELECT LCASE( 'ChoCoLatE' );
```

## LEFT 函数 [String]

从字符串开始处返回多个字符。

## 语法

```
LEFT( string-expression, integer-expression )
```

## 参数

- **string-expression** 字符串。
- **integer-expression** 要返回的字符数。

## 返回值

LONG VARCHAR

LONG NVARCHAR

## 注释

如果字符串包含多字节字符，并且使用了适当的归类，则返回的字节数可能大于指定的字符数。

可以指定比参数字符串表达式中的值大的 *integer-expression*。这种情况下将返回整个值。

如果输入字符串使用字符长度语义，就会在可能的情况下根据字符长度语义对返回值进行说明。

## 另请参见

- [“RIGHT 函数 \[String\]” 一节第 393 页](#)
- [“UltraLite 字符串函数” 一节第 327 页](#)

## 标准和兼容性

- **SQL/2003** 服务商扩充。

## 示例

以下语句返回 Customers 表中每个 Surname 值的前 5 个字符。

```
SELECT LEFT( Surname, 5) FROM Customers;
```

## LENGTH 函数 [String]

返回指定字符串中的字符数。

### 语法

**LENGTH**( *string-expression* )

### 参数

- **string-expression** 字符串。

### 返回值

INT

### 注释

此函数用于确定字符串的长度。例如，为 *string-expression* 指定列名以确定该列中各值的长度。

如果字符串包含多字节字符，并且使用了适当的归类，则 LENGTH 返回字符数而不是字节数。如果字符串数据类型为 BINARY，则 LENGTH 函数的行为将与 BYTE\_LENGTH 函数的行为相同。

#### 注意

对于 CHAR、VARCHAR 和 LONG VARCHAR 数据类型，LENGTH 函数和 CHAR\_LENGTH 函数可以互换使用。不过，对于 BINARY 和位数组数据类型，必须使用 LENGTH 函数。

### 另请参见

- “BYTE\_LENGTH 函数 [String]” 一节第 335 页
- “国际语言和字符集” 《SQL Anywhere 服务器 - 数据库管理》
- “UltraLite 字符串函数” 一节第 327 页

### 标准和兼容性

- **SQL/2003** 服务商扩充。

### 示例

以下语句返回值 9。

```
SELECT LENGTH( 'chocolate' );
```

## LESSER 函数 [Miscellaneous]

返回两个参数值中的较小参数值。

### 语法

**LESSER**( *expression-1*, *expression-2* )

### 参数

- **expression-1** 要比较的第一个参数值。
- **expression-2** 要比较的第二个参数值。

### 返回值

此函数的返回类型取决于指定的表达式。即，数据库服务器计算此函数时，首先搜索可用于比较所有表达式的数据类型。如果找到通用类型，则数据库服务器比较表达式，然后以用于比较的类型返回结果。如果数据库服务器找不到通用比较类型，则返回错误。

### 注释

如果两个参数相等，则返回第一个值。

### 另请参见

- [“GREATER 函数 \[Miscellaneous\]”一节第 361 页](#)

### 标准和兼容性

- **SQL/2003** 服务商扩充。

### 示例

以下语句返回值 5。

```
SELECT LESSER( 10, 5 ) FROM dummy;
```

## LIST 函数 [Aggregate]

返回以逗号分隔的值列表。

### 语法

```
LIST(  
[ DISTINCT ] string-expression  
[ , delimiter-string ] )
```

### 参数

- **string-expression** 字符串表达式，通常是列名。对于列中的每一行，其值都会添加到一个逗号分隔的列表中。如果指定了 DISTINCT，则只添加唯一的值。
- **delimiter-string** 列表项的分隔字符串。缺省设置是逗号。如果设为 NULL 值或空字符串，则没有分隔符。*delimiter-string* 必须是常量。

### 返回值

LONG VARCHAR  
LONG NVARCHAR

## 注释

NULL 值不会添加到列表中。LIST ( X ) 返回一串组中每一行 X 的所有非 NULL 值（带分隔符）。如果组中没有任何一行具有明确的 X 值，则 LIST( X ) 返回空字符串。

LIST 函数不能用作窗口函数，但可用作窗口函数的输入。

## 标准和兼容性

- **SQL/2003** 服务商扩充。

## 示例

以下语句返回 Employees 表中的全部街道地址。

```
SELECT LIST( Street ) FROM Employees;
```

## LOCATE 函数 [String]

返回一个字符串在另一个字符串中的位置。

## 语法

**LOCATE**( *string-expression-1*, *string-expression-2* [, *integer-expression* ] )

## 参数

- **string-expression-1** 被搜索的字符串。
- **string-expression-2** 要搜索的字符串。
- **integer-expression** 字符串中开始进行搜索的字符位置。第一个字符在位置 1。如果起始偏移是负值，则定位函数返回最后一个匹配字符串偏移而非第一个。负的偏移指示从搜索中排除字符串尾的多长一部分。排除的字节数计算公式为  $(-1 * \text{偏移}) - 1$ 。

## 返回值

INT

## 注释

如果指定了 *integer-expression*，则从字符串中的该偏移处开始搜索。

第一个字符串可以是长字符串（长于 255 个字节），但第二个字符串的长度不能超过 255 个字节。如果第二个参数是长字符串，此函数返回 NULL 值。如果未找到字符串，则返回 0。搜索零长度字符串将返回 1。如果有某个参数为 NULL，则结果为 NULL。

如果使用多字节字符，并具有适当的归类，则开始位置和返回值可能不同于字节的位置。

## 另请参见

- [“UltraLite 字符串函数”一节第 327 页](#)
- [“CHARINDEX 函数 \[String\]”一节第 339 页](#)

### 标准和兼容性

- **SQL/2003** 服务商扩充。

### 示例

以下语句返回值 8。

```
SELECT LOCATE(  
    'office party this week - rsvp as soon as possible',  
    'party',  
    2 );
```

## LOG 函数 [Numeric]

返回一个数字的自然对数。

### 语法

**LOG**( *numeric-expression* )

### 参数

- **numeric-expression** 数字。

### 返回值

此函数将其参数转换为 DOUBLE，以双精度浮点执行计算，然后返回 DOUBLE 值作为结果。如果参数为 NULL，则结果为 NULL。

### 注释

参数是返回任何内置数字数据类型值的表达式。

### 另请参见

- [“LOG10 函数 \[Numeric\]” 一节第 373 页](#)

### 标准和兼容性

- **SQL/2003** 服务商扩充。

### 示例

以下语句返回 50 的自然对数。

```
SELECT LOG( 50 );
```

## LOG10 函数 [Numeric]

返回一个数字的以 10 为底的对数。

### 语法

**LOG10**( *numeric-expression* )

### 参数

- **numeric-expression** 数字。

### 返回值

此函数将其参数转换为 DOUBLE，并以双精度浮点执行计算。如果参数为 NULL，则结果为 NULL。

### 注释

参数是返回任何内置数字数据类型值的表达式。

### 另请参见

- [“LOG 函数 \[Numeric\]”一节第 373 页](#)

### 标准和兼容性

- **SQL/2003** 服务商扩充。

### 示例

以下语句返回 50 的以 10 为底的对数。

```
SELECT LOG10( 50 );
```

## LOWER 函数 [String]

将字符串中的所有字符转换成小写形式。此函数与 LCASE 函数完全相同。

### 语法

```
LOWER( string-expression )
```

### 参数

- **string-expression** 要转换的字符串。

### 返回值

CHAR

NCHAR

LONG VARCHAR

VARCHAR

NVARCHAR

### 注释

LCASE 函数与 LOWER 函数完全相同。



### 另请参见

- “LCASE 函数 [String]” 一节第 368 页
- “UCASE 函数 [String]” 一节第 413 页
- “UPPER 函数 [String]” 一节第 414 页
- “UltraLite 字符串函数” 一节第 327 页

### 标准和兼容性

- **SQL/2003** 核心特性。

### 示例

以下语句返回值 chocolate。

```
SELECT LOWER( 'chOCOLate' );
```

## LTRIM 函数 [String]

删除字符串中的前导和尾随空白。

### 语法

```
LTRIM( string-expression )
```

### 参数

- **string-expression** 要剪裁的字符串。

### 返回值

VARCHAR

NVARCHAR

LONG VARCHAR

LONG NVARCHAR

### 注释

结果的实际长度为表达式的长度减去删除的字符数。如果删除了所有字符，则结果为空字符串。

如果参数可以为空值，则结果也可以为空值。

如果参数为空值，则结果为空值。

### 另请参见

- “RTRIM 函数 [String]” 一节第 394 页
- “TRIM 函数 [String]” 一节第 410 页
- “UltraLite 字符串函数” 一节第 327 页

### 标准和兼容性

- **SQL/2003** 服务商扩充。

由 SQL/2003 标准定义的 TRIM 说明（LEADING 和 TRAILING）分别由 SQL Anywhere LTRIM 和 RTRIM 函数提供。

### 示例

以下语句返回值 Test Message，其中的所有前导空白都已删除。

```
SELECT LTRIM( '      Test Message' );
```

## MAX 函数 [Aggregate]

返回在每一组行中找到的最大 *expression* 值。

### 语法 1

```
MAX( expression | DISTINCT expression )
```

### 参数

- **expression** 要计算最大值的表达式。这通常是列名。
- **DISTINCT expression** 返回的值与 MAX( *expression* ) 相同，将其列出是为了保持完整性。

### 返回值

与参数相同的数据类型。

### 注释

忽略 *expression* 为 NULL 的行。对于不包含任何行的组返回 NULL。

### 另请参见

- [“MIN 函数 \[Aggregate\]” 一节第 376 页](#)

### 标准和兼容性

- **SQL/2003** 核心特性。

### 示例

以下语句返回值 138948.000，表示 Employees 表中的最高薪水。

```
SELECT MAX( Salary )  
FROM Employees;
```

## MIN 函数 [Aggregate]

返回在每一组行中找到的最小表达式值。

### 语法 1

```
MIN( expression | DISTINCT expression )
```

### 参数

- **expression** 要计算最小值的表达式。这通常是列名。
- **DISTINCT expression** 返回的值与 `MIN( expression )` 相同，将其列出是为了保持完整性。

### 返回值

与参数相同的数据类型。

### 注释

忽略 `expression` 为 `NULL` 的行。对于不包含任何行的组返回 `NULL`。

### 另请参见

- “MAX 函数 [Aggregate]” 一节第 376 页

### 标准和兼容性

- **SQL/2003** 核心特性。

### 示例

以下语句返回值 24903.000，表示 `Employees` 表中的最低薪水。

```
SELECT MIN( Salary )  
FROM Employees;
```

## MINUTE 函数 [Date and time]

返回日期时间值的分钟部分。

### 语法

**MINUTE**( *datetime-expression* )

### 参数

- **datetime-expression** 日期时间值。

### 返回值

`SMALLINT`

### 注释

返回的值是 0 到 59 之间的某个数字，对应于日期时间的分钟。

### 标准和兼容性

- **SQL/2003** 服务商扩充。

### 示例

以下语句返回值 22。

```
SELECT MINUTE( '1998-07-13 12:22:34' );
```

## MINUTES 函数 [Date and time]

所指定的信息会更改此函数的行为:

- 如果提供一个日期, 此函数返回自 0000-02-29 以来的分钟数。

### 注意

0000-02-29 并不表示实际日期, 它是日期算法使用的日期。

- 如果提供两个时间戳, 此函数会返回它们之间的整分钟数。替代方法为使用 DATEDIFF 函数。
- 如果提供一个日期和一个整数, 此函数会为指定时间戳添加整数分钟数。替代方法为使用 DATEADD 函数。

### 语法 1: integer

```
MINUTES( [ datetime-expression, ] datetime-expression )
```

### 语法 2: timestamp

```
MINUTES( datetime-expression, integer-expression )
```

### 参数

- **datetime-expression** 日期和时间。
- **integer-expression** 要添加到 *datetime-expression* 中的分钟数。如果 *integer-expression* 是负数, 则从日期时间值中减去相应的秒数。如果提供整数表达式, 则必须将 *datetime-expression* 显式地转换为 DATETIME 数据类型。

### 返回值

INT

TIMESTAMP

### 注释

由于此函数返回整数, 因此当使用语法 1 并且时间戳大于或等于 4083-03-23 02:08:00 时可能会发生溢出。

### 另请参见

- [“CAST 函数 \[Data type conversion\]” 一节第 337 页](#)

### 标准和兼容性

- **SQL/2003** 服务商扩充。

## 示例

以下语句返回值 240，表示第二个时间戳比第一个时间戳晚 240 分钟。建议使用第二个示例 (DATEDIFF)。

```
SELECT MINUTES( '1999-07-13 06:07:12',
                '1999-07-13 10:07:12' );

SELECT DATEDIFF( minute,
                '1999-07-13 06:07:12',
                '1999-07-13 10:07:12' );
```

以下语句返回值 1051040527。

```
SELECT MINUTES( '1998-07-13 06:07:12' );
```

以下语句返回时间戳 12.05.99 21:10:070.000。建议使用第二个示例 (DATEADD)。

```
SELECT MINUTES( CAST( '1999-05-12 21:05:07'
                    AS DATETIME ), 5);

SELECT DATEADD( minute, 5, '1999-05-12 21:05:07' );
```

## ML\_GET\_SERVER\_NOTIFICATION [系统]

此函数允许 Ultralite 用户使用轻量级轮询，来查询 MobiLink 服务器上服务器启动的同步请求的通告程序。

### 语法

```
ML_GET_SERVER_NOTIFICATION(notifier, address, key)
```

### 参数

- **Notifier** MobiLink 服务器上要轮询的通告程序的名称。
- **地址** 流参数，以下面的格式指定：

```
tcpip{host=pc1;port=1234}
```

示例如上。

- **键** 可选。通知请求关键字。

### 返回值

返回给定请求关键字的通知请求的主题和内容。

### 注释

如果给定的请求关键字没有任何请求，或者 MobiLink 服务器中不存在此通告程序的名称，则结果为 NULL。如果为请求关键字提供的是 NULL，则会将用户的远程 ID 用作请求关键字。如果不存在请求，则采用以下格式返回结果消息：[subject]content（例如，[sync]profile1）。

此函数在检索来自 MobiLink 服务器的响应时，通过网络进行通信。因此，由于需要网络等待，此函数可能需要较长的执行时间。在执行期间，可能存在多个时间段，在这些时间段内，函数可以在后台执行，同时允许在运行时对其它连接执行工作。然而，这些时间段无法得到保证，而且取决

于 SQL 的复杂性。对用户而言，在检索要在此函数中使用的 MobiLink 地址时，建议的方法是配合使用 `sync_profile_option_value` 函数与现有的同步配置文件，以获取 **Stream** 配置文件选项的值。此函数调用所返回的值可直接用作 MobiLink 地址参数。

#### 另请参见

- [“SYNC\\_PROFILE\\_OPTION\\_VALUE 函数 \[系统\]” 一节第 408 页](#)

#### 标准和兼容性

#### 示例

```
Select ml_get_server_notification('Notifier1',  
    'tcpip{host=sybase;port=1234}', 'MyKey')
```

## MOD 函数 [Numeric]

返回一个整数除以另一个整数之后产生的余数。

#### 语法

**MOD**( *dividend*, *divisor* )

#### 参数

- **dividend** 被除数，即除法的分子。
- **divisor** 除数，即除法的分母。

#### 返回值

SMALLINT

INT

NUMERIC

#### 注释

被除数为负数时结果为负数或零。除数的符号没有影响。

#### 另请参见

- [“REMAINDER 函数 \[Numeric\]” 一节第 389 页](#)

#### 标准和兼容性

- **SQL/2003** 核心 SQL 之外的 SQL 基础特性。

#### 示例

以下语句返回值 2。

```
SELECT MOD( 5, 3 );
```

## MONTH 函数 [Date and time]

返回给定日期的月份。

### 语法

**MONTH**( *date-expression* )

### 参数

- **date-expression** 日期时间值。

### 返回值

SMALLINT

### 注释

返回的值是 1 到 12 之间的某个数字，对应于日期时间的月份。

### 标准和兼容性

- **SQL/2003** 服务商扩充。

### 示例

以下语句返回值 7。

```
SELECT MONTH( '1998-07-13' );
```

## MONTHNAME 函数 [Date and time]

返回日期中的月份名称。

### 语法

**MONTHNAME**( *date-expression* )

### 参数

- **date-expression** 日期时间值。

### 返回值

VARCHAR

### 注释

MONTHNAME 函数返回字符串，即使结果是数字类型（如表示二月的 2）。

### 标准和兼容性

- **SQL/2003** 服务商扩充。

## 示例

以下语句返回值 September。

```
SELECT MONTHNAME ( '1998-09-05' );
```

## MONTHS 函数 [Date and time]

所指定的信息会更改此函数的行为：

- 如果提供一个日期，此函数返回自 0000-02 以来的月份数。

### 注意

0000-02 并不表示实际日期，它是日期算法使用的日期。

- 如果提供两个时间戳，此函数会返回它们之间的整月份数。替代方法为使用 DATEDIFF 函数。
- 如果提供一个日期和一个整数，此函数会为指定时间戳添加整数分钟数。替代方法为使用 DATEADD 函数。

### 语法 1: integer

```
MONTHS( [ datetime-expression, ] datetime-expression )
```

### 语法 2: timestamp

```
MONTHS( datetime-expression, integer-expression )
```

### 参数

- **datetime-expression** 日期和时间。
- **integer-expression** 要添加到 *datetime-expression* 中的月份数。如果 *integer-expression* 是负数，则从日期时间值中减去相应的月数。如果提供 *integer-expression*，则必须将 *datetime-expression* 显式地转换为日期时间数据类型。

有关转换数据类型的信息，请参见“CAST 函数 [Data type conversion]”一节第 337 页。

### 返回值

INT

TIMESTAMP

### 注释

MONTHS 的值是根据两个日期之间每个月的 1 号的数目计算出的。

### 标准和兼容性

- **SQL/2003** 服务商扩充。

### 示例

以下语句返回值 2，表示第二个日期比第一个日期晚两个月。建议使用第二个示例 (DATEDIFF)。



```
SELECT MONTHS( '1999-07-13 06:07:12',  
              '1999-09-13 10:07:12' );
```

```
SELECT DATEDIFF( month,  
              '1999-07-13 06:07:12',  
              '1999-09-13 10:07:12' );
```

以下语句返回值 23981。

```
SELECT MONTHS( '1998-07-13 06:07:12' );
```

以下语句返回时间戳 1999-10-12 21:05:070.000。建议使用第二个示例 (DATEADD)。

```
SELECT MONTHS( CAST( '1999-05-12 21:05:07'  
                  AS DATETIME ), 5);  
  
SELECT DATEADD( month, 5, '1999-05-12 21:05:07' );
```

## NEWID 函数 [Miscellaneous]

生成 UUID（通用唯一标识符）值。UUID 与 GUID（全局唯一标识符）相同。

### 语法

```
NEWID( )
```

### 参数

没有任何与 NEWID 函数关联的参数。

### 返回值

UNIQUEIDENTIFIER

### 注释

NEWID 函数可以在列的 DEFAULT 子句中使用。

UUID 可用于唯一地标识表中的行。在一台计算机上生成的值与在另一台计算机上生成的值不匹配，因此，在同步和复制环境中可将这些值作为关键字来使用。

### 另请参见

- “NEWID 缺省值”一节 《SQL Anywhere 服务器 - SQL 的用法》
- “STRTOUUID 函数 [String]”一节第 404 页
- “UIDTOSTR 函数 [String]”一节第 416 页

### 标准和兼容性

- **SQL/2003** 服务商扩充。

### 示例

以下语句创建一个包含两列、名为 mytab 的表。列 pk 的数据类型是唯一标识符，它指派 NEWID 函数作为缺省值。列 c1 的数据类型是整型。

```
CREATE TABLE mytab(  
    pk UNIQUEIDENTIFIER PRIMARY KEY DEFAULT NEWID(),  
    c1 INT );
```

以下语句返回字符串形式的唯一标识符:

```
SELECT NEWID();
```

例如, 返回的值可能是 96603324-6FF6-49DE-BF7D-F44C1C7E6856。

## NOW 函数 [Date and time]

返回当前的年、月、日、小时、分钟、秒和秒的小数部分。其准确性受系统时钟准确性的限制。

### 语法

```
NOW( * )
```

### 返回值

TIMESTAMP

### 注释

NOW 函数返回的信息与 GETDATE 函数返回的信息及 CURRENT\_TIMESTAMP 特殊值相同。

### 另请参见

- [“GETDATE 函数 \[Date and time\]” 一节第 361 页](#)
- [“CURRENT\\_TIMESTAMP 特殊值” 一节第 287 页](#)

### 标准和兼容性

- **SQL/2003** 服务商扩充。

### 示例

以下语句返回当前日期和时间。

```
SELECT NOW( * );
```

## NULLIF 函数 [Miscellaneous]

通过比较表达式提供缩写的 CASE 表达式。

### 语法

```
NULLIF( expression-1, expression-2 )
```

### 参数

- **expression-1** 要比较的表达式。
- **expression-2** 要比较的表达式。

## 返回值

第一个参数的数据类型。

## 注释

NULLIF 比较两个表达式的值。

如果第一个表达式的值与第二个表达式的值相等，NULLIF 返回 NULL。

如果第一个表达式的值不等于第二个表达式的值，或者第二个表达式为 NULL，则 NULLIF 返回第一个表达式。

NULLIF 函数提供了编写某些 CASE 表达式的简便方法。

## 另请参见

- [“CASE 表达式”一节第 305 页](#)

## 标准和兼容性

- **SQL/2003** 核心特性。

## 示例

以下语句返回值 a:

```
SELECT NULLIF( 'a', 'b' );
```

下面的语句返回 NULL。

```
SELECT NULLIF( 'a', 'a' );
```

## SQL 函数 (P-Z)

列出了每个函数，并在它旁边指出函数类型（数字、字符等等）。

有关指向给定类型所有函数的链接，请参见“[函数类型](#)”一节第 324 页。

### 另请参见

- “[SQL 函数 \(A-D\)](#)”一节 《[SQL Anywhere 服务器 - SQL 参考](#)》
- “[SQL 函数 \(E-O\)](#)”一节 《[SQL Anywhere 服务器 - SQL 参考](#)》

## PATINDEX 函数 [String]

返回一个整数，该整数表示模式第一次在字符串中出现的起始位置。

### 语法

**PATINDEX**( '%*pattern*%', *string-expression* )

### 参数

- **pattern** 要搜索的模式。如果省略前导百分号通配符，则模式出现在字符串起始处时，PATINDEX 函数返回一 (1)，否则返回零。

UltraLite 的模式使用以下通配符：

通配符	匹配项
_ (下划线)	任意一个字符
% (百分号)	包含零个或多个字符的任意字符串
[]	指定范围或集合中的任何单个字符
[^]	不在指定范围或集合中的任何单个字符

- **string-expression** 要在其中搜索模式的字符串。

### 返回值

INT

### 注释

PATINDEX 函数返回模式第一次出现的开始位置。如果未找到模式，则返回零 (0)。

### 另请参见

- “[LOCATE 函数 \[String\]](#)”一节第 372 页
- “[UltraLite 字符串函数](#)”一节第 327 页

### 标准和兼容性

- **SQL/2003** 服务商扩充。

### 示例

以下语句返回值 2。

```
SELECT PATINDEX( '%hoco%', 'chocolate' );
```

以下语句返回值 11。

```
SELECT PATINDEX( '%4_5_', '0a1A 2a3A 4a5A' );
```

以下语句返回 14，即字符串表达式中第一个非字母数字字符。注意，如果数据库区分大小写，则可使用模式 '%[^a-z0-9]%' 代替 '%[^a-zA-Z0-9]%'。

```
SELECT PATINDEX( '%[^a-zA-Z0-9]%', 'SQLAnywhere11 has many new features' );
```

要获得字符串中的第一个字母数字单词，可以通过类似下面的方式：

```
SELECT LEFT( @string, PATINDEX( '%[^a-zA-Z0-9]%', @string ) );
```

## PI 函数 [Numeric]

返回数字值 PI。

### 语法

```
PI( * )
```

### 返回值

DOUBLE

### 标准和兼容性

- **SQL/2003** 服务商扩充。

### 注释

此函数返回一个 DOUBLE 值。

### 示例

以下语句返回值 3.141592653……

```
SELECT PI( * );
```

## POWER 函数 [Numeric]

以一个数字为底数另一个数字为指数计算乘方值。

**语法**

**POWER**( *numeric-expression-1*, *numeric-expression-2* )

**参数**

- **numeric-expression-1** 底数。
- **numeric-expression-2** 指数。

**返回值**

DOUBLE

**注释**

此函数将其参数转换为 DOUBLE，并以双精度浮点执行计算。如果任何参数为 NULL，则结果为 NULL 值。

**标准和兼容性**

- **SQL/2003** 服务商扩充。

**示例**

以下语句返回值 64。

```
SELECT POWER( 2, 6 );
```

## QUARTER 函数 [Date and time]

从提供的日期表达式中返回一个表示某季度的数字。

**语法**

**QUARTER**( *date-expression* )

**参数**

- **date-expression** 日期。

**返回值**

INT

**注释**

有如下这些季度：

Quarter	期限（含起止日期）
1	1 月 1 日到 3 月 31 日
2	4 月 1 日到 6 月 30 日

Quarter	期限 (含起止日期)
3	7 月 1 日到 9 月 30 日
4	10 月 1 日到 12 月 31 日

### 标准和兼容性

- **SQL/2003** 服务商扩充。

### 示例

以下语句返回值 2。

```
SELECT QUARTER( '1987/05/02' );
```

## RADIANS 函数 [Numeric]

将数字由角度转换成弧度。

### 语法

**RADIANS**( *numeric-expression* )

### 参数

- **numeric-expression** 数字（以度为单位）。此角度将转换成弧度。

### 返回值

DOUBLE

### 注释

此函数将其参数转换为 DOUBLE，并以双精度浮点执行计算。

### 标准和兼容性

- **SQL/2003** 服务商扩充。

### 示例

以下语句返回近似值 0.5236。

```
SELECT RADIANS( 30 );
```

## REMAINDER 函数 [Numeric]

返回一个整数除以另一个整数之后产生的余数。

### 语法

**REMAINDER**( *dividend*, *divisor* )

### 参数

- **dividend** 被除数，即除法的分子。
- **divisor** 除数，即除法的分母。

### 返回值

INTEGER  
NUMERIC

### 注释

可以尝试使用 MOD 函数作为替代。

### 另请参见

- [“MOD 函数 \[Numeric\]” 一节第 380 页](#)

### 标准和兼容性

- **SQL/2003** 服务商扩充。

### 示例

以下语句返回值 2。

```
SELECT REMAINDER( 5, 3 );
```

## REPEAT 函数 [String]

以指定的次数连接字符串。

### 语法

**REPEAT**( *string-expression*, *integer-expression* )

### 参数

- **string-expression** 要重复的字符串。
- **integer-expression** 字符串的重复次数。如果 *integer-expression* 是负数，则返回空字符串。

### 返回值

LONG VARCHAR  
LONG NVARCHAR



## 注释

如果结果字符串的实际长度超过返回类型的最大值，会产生错误。结果将截断为所允许的最大字符串大小。

可以尝试使用 REPLICATE 函数作为替代。

## 另请参见

- “REPLICATE 函数 [String]” 一节第 392 页
- “UltraLite 字符串函数” 一节第 327 页

## 标准和兼容性

- **SQL/2003** 服务商扩充。

## 示例

下面的语句返回值 repeatrepeatrepeat。

```
SELECT REPEAT( 'repeat', 3 );
```

# REPLACE 函数 [String]

使用另一个字符串替换某个字符串，并返回新的结果。

## 语法

```
REPLACE( original-string, search-string, replace-string )
```

## 参数

如果有某个参数为 NULL，此函数返回 NULL。

- **original-string** 被搜索的字符串。可为任意长度。
- **search-string** 要进行搜索并以 *replace-string* 进行替换的字符串。此字符串的长度不应超过 255 个字节。如果 *search-string* 是空字符串，则按原样返回原始字符串。
- **replace-string** 替代字符串，用于替换 *search-string*。可为任意长度。如果 *replacement-string* 是空字符串，则会删除出现的所有 *search-string*。

## 返回值

LONG VARCHAR

LONG NVARCHAR

## 注释

此函数替换所有出现的字符串。

对于区分大小写的数据库，比较也区分大小写。

### 另请参见

- “SUBSTRING 函数 [String]” 一节第 405 页
- “CHARINDEX 函数 [String]” 一节第 339 页
- “UltraLite 字符串函数” 一节第 327 页

### 标准和兼容性

- **SQL/2003** 服务商扩充。

### 示例

以下语句返回值 xx.def.xx.ghi。

```
SELECT REPLACE( 'abc.def.abc.ghi', 'abc', 'xx' );
```

以下语句生成包含 ALTER PROCEDURE 语句的结果集，这些语句在执行时会修复那些引用已重命名的表的存储过程。（表名必须唯一才会有用处。）

```
SELECT REPLACE(
    REPLACE( proc_defn, 'OldTableName', 'NewTableName' ),
    'CREATE PROCEDURE',
    'ALTER PROCEDURE')
FROM SYS.SYSPROCEDURE
WHERE proc_defn LIKE '%OldTableName%';
```

## REPLICATE 函数 [String]

以指定的次数连接字符串。

### 语法

```
REPLICATE( string-expression, integer-expression )
```

### 参数

- **string-expression** 要重复的字符串。
- **integer-expression** 字符串的重复次数。

### 返回值

LONG VARCHAR

LONG NVARCHAR

### 注释

如果结果字符串的实际长度超过返回类型的最大值，会产生错误。结果将截断为所允许的最大字符串大小。

可以尝试使用 REPEAT 函数作为替代。

### 另请参见

- “REPEAT 函数 [String]” 一节第 390 页
- “UltraLite 字符串函数” 一节第 327 页

### 标准和兼容性

- **SQL/2003** 服务商扩充。

### 示例

下面的语句返回值 repeatrepeatrepeat。

```
SELECT REPLICATE( 'repeat', 3 );
```

## RIGHT 函数 [String]

返回字符串中最右边的字符。

### 语法

```
RIGHT( string-expression, integer-expression )
```

### 参数

- **string-expression** 从左边截断的字符串。
- **integer-expression** 要返回的从字符串结尾处开始计数的字符数。

### 返回值

LONG VARCHAR

LONG NVARCHAR

### 注释

如果字符串包含多字节字符，并且使用了适当的归类，则返回的字节数可能大于指定的字符数。指定的 *integer-expression* 可以比列中的值大。这种情况下将返回整个值。

如果输入字符串使用字符长度语义，就会在可能的情况下根据字符长度语义对返回值进行说明。

### 另请参见

- “LEFT 函数 [String]” 一节第 369 页
- “国际语言和字符集” 《SQL Anywhere 服务器 - 数据库管理》
- “UltraLite 字符串函数” 一节第 327 页

### 标准和兼容性

- **SQL/2003** 服务商扩充。

### 示例

以下语句返回 Customers 表中每个 Surname 值的最后 5 个字符。

```
SELECT RIGHT( Surname, 5) FROM Customers;
```

## ROUND 函数 [Numeric]

将 *numeric-expression* 舍入到小数点后的指定 *integer-expression* 位数。

### 语法

```
ROUND( numeric-expression, integer-expression )
```

### 参数

- **numeric-expression** 要舍入的数字（传递给函数）。
- **integer-expression** 正整数指定小数点右边舍入到的有效位数。负数表达式指定小数点左边舍入到的有效位数。

### 返回值

NUMERIC

### 注释

此函数的结果为数字或双精度值。当结果为数字值并且整数 *integer-expression* 是负值时，精度将加一。

### 另请参见

- [“TRUNCNUM 函数 \[Numeric\]” 一节第 411 页](#)

### 标准和兼容性

- **SQL/2003** 服务商扩充。

### 示例

以下语句返回值 123.200。

```
SELECT ROUND( 123.234, 1 );
```

## RTRIM 函数 [String]

删除字符串中的前导和尾随空白。

### 语法

```
RTRIM( string-expression )
```

### 参数

- **string-expression** 要剪裁的字符串。

**返回值**

VARCHAR  
NVARCHAR  
LONG VARCHAR  
LONG NVARCHAR

**注释**

结果的实际长度为表达式的长度减去删除的字符数。如果删除了所有字符，则结果为空字符串。  
如果参数为空值，则结果为空值。

**另请参见**

- [“TRIM 函数 \[String\]”一节第 410 页](#)
- [“LTRIM 函数 \[String\]”一节第 375 页](#)
- [“UltraLite 字符串函数”一节第 327 页](#)

**标准和兼容性**

- **SQL/2003** 服务商扩充。  
由 SQL/2003 标准定义的 TRIM 说明（LEADING 和 TRAILING）分别由 SQL Anywhere LTRIM 和 RTRIM 函数提供。

**示例**

以下语句返回字符串 Test Message，其中的所有末尾空格都已删除。

```
SELECT RTRIM( 'Test Message      ' );
```

## SECOND 函数 [Date and time]

返回给定日期的秒。

**语法**

```
SECOND( datetime-expression )
```

**参数**

- **datetime-expression** 日期时间值。

**返回值**

SMALLINT

**注释**

返回 0 到 59 之间的一个数字，该数字对应于给定日期时间值的秒部分。

## 标准和兼容性

- **SQL/2003** 服务商扩充。

## 示例

以下语句返回值 25。

```
SELECT SECOND( '1998-07-13 21:21:25' );
```

## SECONDS 函数 [Date and time]

此函数的行为会视所提供内容的不同而变化：

- 如果提供一个日期，此函数返回自 0000-02-29 以来的秒数。

### 注意

0000-02-29 并不表示实际日期，它是日期算法使用的日期。

- 如果提供两个时间戳，此函数返回它们之间的整秒数。替代方法为使用 DATEDIFF 函数。
- 如果提供一个日期和一个整数，此函数会为指定时间戳添加整数秒数。替代方法为使用 DATEADD 函数。

## 语法 1: integer

```
SECONDS( [ datetime-expression, ] datetime-expression )
```

## 语法 2: timestamp

```
SECONDS( datetime-expression, integer-expression )
```

## 参数

- **datetime-expression** 日期和时间。
- **integer-expression** 要添加到 *datetime-expression* 中的秒数。如果 *integer-expression* 是负数，则从日期时间值中减去相应的秒数。如果提供整数表达式，则必须将 *datetime-expression* 显式地转换为日期时间数据类型。

## 返回值

INTEGER

TIMESTAMP

## 另请参见

- “CAST 函数 [Data type conversion]” 一节第 337 页
- “DATEADD 函数 [Date and time]” 一节第 348 页
- “DATEDIFF 函数 [Date and time]” 一节第 349 页

## 标准和兼容性

- **SQL/2003** 服务商扩充。

## 示例

以下语句返回值 14400，表示第二个时间戳比第一个时间戳晚 14400 秒。

```
SELECT SECONDS( '1999-07-13 06:07:12',
               '1999-07-13 10:07:12' );

SELECT DATEDIFF( second,
               '1999-07-13 06:07:12',
               '1999-07-13 10:07:12' );
```

以下语句返回值 63062431632。

```
SELECT SECONDS( '1998-07-13 06:07:12' );
```

以下语句返回日期时间 1999-05-12 21:05:120.000。

```
SELECT SECONDS( CAST( '1999-05-12 21:05:07'
                    AS TIMESTAMP ), 5);

SELECT DATEADD( second, 5, '1999-05-12 21:05:07' );
```

## SHORT\_PLAN 函数 [Miscellaneous]

以字符串形式返回 SQL 语句的 UltraLite 计划优化策略的简短说明。此说明与 EXPLANATION 函数返回的说明相同。

## 语法

```
SHORT_PLAN( string-expression )
```

## 注释

对于某些查询，UltraLite 的执行计划可能不同于为 SQL Anywhere 选择的计划。

## 参数

- **string-expression** SQL 语句，通常是 SELECT 语句，但也可以是 UPDATE 或 DELETE 语句。

## 返回值

LONG VARCHAR

## 另请参见

- [“EXPLANATION 函数 \[Miscellaneous\]”](#) 一节第 359 页

## 标准和兼容性

- **SQL/2003** 服务商扩充。

## 示例

下面的语句将 SELECT 语句作为字符串参数传递，并返回查询的执行计划。

```
SELECT SHORT_PLAN(  
    'SELECT * FROM Departments WHERE DepartmentID > 100' );
```

此信息有助于决定要添加的索引或数据库的构建方法，以便获得更好的性能。

## SIGN 函数 [Numeric]

返回一个数字的符号。

### 语法

```
SIGN( numeric-expression )
```

### 参数

- **numeric-expression** 要返回其符号的数字。

### 返回值

SMALL INT

### 注释

对于负数，SIGN 函数返回 -1。

对于零，SIGN 函数返回 0。

对于正数，SIGN 函数返回 1。

### 标准和兼容性

- **SQL/2003** 服务商扩充。

### 示例

以下语句返回值 -1

```
SELECT SIGN( -550 );
```

## SIMILAR 函数 [String]

返回一个数字，以表示两个字符串之间的相似性。

### 语法

```
SIMILAR( string-expression-1, string-expression-2 )
```

### 参数

- **string-expression-1** 要比较的第一个字符串。



- **string-expression-2** 要比较的第二个字符串。

### 返回值

SMALL INT

### 注释

此函数返回 0 和 100 之间的一个整数，表示两个字符串之间的相似性。结果可以被解释成两个字符串之间匹配字符的百分比。值 100 表示两个字符串完全相同。

此函数可用于更正名称（如客户）列表。某些客户可能用稍微不同的名称多次添加到列表中。将表与自身连接，并生成一个报告，其中列出所有大于 90% 但小于 100% 的相似性。

为 SIMILAR 函数执行的计算要比仅计算匹配的字符数更为复杂。

### 另请参见

- [“UltraLite 字符串函数”一节第 327 页](#)

### 标准和兼容性

- **SQL/2003** 服务商扩充。

### 示例

以下语句返回值 75，表示两个值的相似性为 75%。

```
SELECT SIMILAR( 'toast', 'coast' );
```

## SIN 函数 [Numeric]

返回一个数字的正弦值。

### 语法

**SIN**( *numeric-expression* )

### 参数

- **numeric-expression** 角度（以弧度为单位）。

### 返回值

DOUBLE

### 注释

SIN 函数返回参数的正弦，参数是以弧度表示的角度。SIN 和 ASIN 函数互为逆运算。

此函数将其参数转换为 DOUBLE，以双精度浮点执行计算，然后返回 DOUBLE 值作为结果。

**另请参见**

- “ASIN 函数 [Numeric]” 一节第 332 页
- “COS 函数 [Numeric]” 一节第 344 页
- “COT 函数 [Numeric]” 一节第 345 页
- “TAN 函数 [Numeric]” 一节第 409 页

**标准和兼容性**

- **SQL/2003** 服务商扩充。

**示例**

以下语句返回 0.52 的 SIN 值。

```
SELECT SIN( 0.52 );
```

## SOUNDEX 函数 [String]

返回表示字符串声音的数字。

**语法**

```
SOUNDEX( string-expression )
```

**参数**

- **string-expression** 要计算的字符串。

**返回值**

SMALLINT

**注释**

字符串的 SOUNDEX 函数值基于第一个字母和随后三个除 H、Y 和 W 之外的辅音。*string-expression* 中的元音将被忽略，除非它们是字符串的第一个字母。重复的字母按一个字母计数。例如，词语 apples 基于字母 A、P、L 和 S。

SOUNDEX 函数会忽略多字节字符。

尽管并不完美，但对于那些发音相似及以相同字母开头的单词，SOUNDEX 函数返回的数字通常是相同的。

SOUNDEX 函数对英语单词的效果最好。对其它语言单词的效果较差。

**另请参见**

- “UltraLite 字符串函数” 一节第 327 页

**标准和兼容性**

- **SQL/2003** 服务商扩充。

### 示例

以下语句返回两个完全相同的数字 3827，表示每个名称的声音。

```
SELECT SOUNDEX( 'Smith' ), SOUNDEX( 'Smythe' );
```

## SPACE 函数 [String]

返回指定数目的空格。

### 语法

```
SPACE( integer-expression )
```

### 参数

- **integer-expression** 要返回的空格数。

### 返回值

LONG VARCHAR

### 注释

如果 *integer-expression* 为负，则返回空字符串。

### 另请参见

- [“UltraLite 字符串函数”一节第 327 页](#)

### 标准和兼容性

- **SQL/2003** 服务商扩充。

### 示例

以下语句返回包含 10 个空格的字符串。

```
SELECT SPACE( 10 );
```

## SQRT 函数 [Numeric]

返回一个数字的平方根。

### 语法

```
SQRT( numeric-expression )
```

### 参数

- **numeric-expression** 要计算其平方根的数字。

### 返回值

DOUBLE

### 注释

此函数将其参数转换为 DOUBLE，以双精度浮点执行计算，然后返回 DOUBLE 值作为结果。

### 标准和兼容性

- **SQL/2003** 服务商扩充。

### 示例

以下语句返回值 3。

```
SELECT SQRT( 9 );
```

## STR 函数 [String]

返回数字的等效字符串。

### 语法

```
STR( numeric-expression [, length [, decimal ] ] )
```

### 参数

- **numeric-expression** 任何 -1E126 与 1E127 之间的近似数字（浮点、实数或双精度）表达式。
- **length** 要返回的字符数（包括小数点、小数点右侧和左侧的所有位以及空白）。缺省值为 10。
- **decimal** 要返回的小数位。缺省值为 0。

### 返回值

VARCHAR

### 注释

如果数字的整数部分超过了指定的长度，则结果是具有指定长度但只包含星号的字符串。例如，以下语句返回 \*\*\*。

```
SELECT STR( 1234.56, 3 );
```

#### 注意

所支持的最大长度为 128。任何不在 1 到 128 范围内的长度都会生成 NULL 结果。

### 另请参见

- [“UltraLite 字符串函数”一节第 327 页](#)

### 标准和兼容性

- **SQL/2003** 服务商扩充。

## 示例

以下语句返回的字符串共有十个字符，前面是六个空格，后面是 1235。

```
SELECT STR( 1234.56 );
```

以下语句返回结果 1234.6。

```
SELECT STR( 1234.56, 6, 1 );
```

## STRING 函数 [String]

将一个或多个字符串连接为一个大字符串。

### 语法

```
STRING( string-expression [, ... ])
```

### 参数

- **string-expression** 要计算的字符串。

如果只提供一个参数，则会将它转换为单个表达式。如果提供多个参数，则将它们连接成单个字符串。

### 返回值

LONG VARCHAR

LONG NVARCHAR

LONG BINARY

### 注释

数字或日期参数在连接前将会转换为字符串。STRING 函数还可用于通过将任何单个表达式作为唯一的参数进行提供来将该表达式转换为字符串。

如果所有参数都为 NULL，则 STRING 返回 NULL。如果任何参数为非 NULL，则将所有 NULL 参数都视为空字符串。

### 另请参见

- [“UltraLite 字符串函数”一节第 327 页](#)

### 标准和兼容性

- **SQL/2003** 服务商扩充。

## 示例

以下语句返回值 *testing123*。

```
SELECT STRING( 'testing', NULL, 123 );
```

## STRTOUUID 函数 [String]

将字符串值转换为唯一标识符（UUID 或 GUID）值。

### 较新的数据库中不需要

在使用 9.0.2 以前的版本创建的数据库中，需要使用 STRTOUUID 和 UUIDTOSTR 函数在 UUID 值的二进制与字符串表示之间进行转换。

在使用 9.0.2 或更高版本创建的数据库中，UNIQUEIDENTIFIER 数据类型已更改为本地数据类型，。在这些版本中不需要使用 STRTOUUID 和 UUIDTOSTR 函数。

有关详细信息，请参见“UltraLite 中的数据类型”一节第 290 页。

### 语法

**STRTOUUID**( *string-expression* )

### 参数

- **string-expression** xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx 格式的字符串。

### 返回值

UNIQUEIDENTIFIER

### 注释

将 xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx（其中 *x* 是十六进制数字）格式的字符串转换为一个唯一的标识符值。

此函数用于在数据库中插入 UUID 值。

### 另请参见

- “UUIDTOSTR 函数 [String]”一节第 416 页
- “NEWID 函数 [Miscellaneous]”一节第 383 页
- “UltraLite 字符串函数”一节第 327 页

### 标准和兼容性

- **SQL/2003** 服务商扩充。

## STUFF 函数 [String]

从一个字符串中删除多个字符，并用另一个字符串替换它们。

### 语法

**STUFF**( *string-expression-1*, *start*, *length*, *string-expression-2* )

### 参数

- **string-expression-1** 要用 STUFF 函数修改的字符串。

- **start** 开始删除字符的字符位置。字符串中第一个字符的位置是 1。
- **length** 要删除的字符数。
- **string-expression-2** 要插入的字符串。要使用 STUFF 函数删除字符串的一部分，请使用 NULL 作为替换字符串。

### 返回值

LONG NVARCHAR

### 另请参见

- “INSERTSTR 函数 [String]” 一节第 366 页
- “UltraLite 字符串函数” 一节第 327 页

### 标准和兼容性

- **SQL/2003** 服务商扩充。

### 示例

以下语句返回值 chocolate pie。

```
SELECT STUFF( 'chocolate cake', 11, 4, 'pie' );
```

## SUBSTRING 函数 [String]

返回字符串的子串。

### 语法

```
{ SUBSTRING | SUBSTR } ( string-expression, start  
[, length ])
```

### 参数

- **string-expression** 从中返回子串的字符串。
- **start** 要返回的子串的开始位置（以字符为单位）。
- **length** 要返回的子串的长度（以字符为单位）。如果指定了 *length*，便将子串限制为不能超过该长度。

### 返回值

LONG BINARY

LONG VARCHAR

LONG NVARCHAR

### 注释

在 UltraLite 中，数据库不具有 `ansi_substring` 选项，但 SUBSTR 函数的行为方式与将 `ansi_substring` 缺省设置为 `on` 一样。此函数的行为与 ANSI/ISO SQL/2003 的行为相同：

- **Start 值** 字符串中的第一个字符处于位置 1。对负或零起始偏移的处理方法是：将其当作在字符串左侧填充了非字符。
- **Length 值** 正的 *length* 指定子串在开始位置右侧第 *length* 个字符处结束。  
负的 *length* 会返回错误。  
*length* 为零时会返回空字符串。

如果 *string-expression* 是二进制数据类型，则 SUBSTRING 函数的行为与 BYTE\_SUBSTR 相同。  
要获取字符串尾端的字符，应使用 RIGHT 函数。

如果输入字符串使用字符长度语义，就会在可能的情况下根据字符长度语义对返回值进行说明。

#### 另请参见

- “BYTE\_SUBSTR 函数 [String]” 一节第 336 页
- “LEFT 函数 [String]” 一节第 369 页
- “RIGHT 函数 [String]” 一节第 393 页
- “CHARINDEX 函数 [String]” 一节第 339 页
- “UltraLite 字符串函数” 一节第 327 页

#### 标准和兼容性

- SQL/2003 核心特性。

#### 示例

下表显示了 SUBSTRING 函数返回的值。

示例	结果
SUBSTRING( 'front yard', 1, 4 )	fron
SUBSTRING( 'back yard', 6, 4 )	yard
SUBSTR( 'abcdefgh', 0, -2 )	返回错误
SUBSTR( 'abcdefgh', -2, 2 )	返回空字符串

## SUM 函数 [Aggregate]

返回每一组行的指定表达式总数。

#### 语法 1

**SUM**( *expression* | **DISTINCT** *expression* )

#### 参数

- **expression** 要求和的对象。这通常是列名。



- **DISTINCT expression** 计算输入中 *expression* 的唯一值的总和。

### 返回值

INTEGER  
DOUBLE  
NUMERIC

### 注释

不包括指定表达式为 NULL 的行。  
对于不包含任何行的组返回 NULL。

### 另请参见

- [“COUNT 函数 \[Aggregate\]” 一节第 345 页](#)
- [“AVG 函数 \[Aggregate\]” 一节第 335 页](#)

### 标准和兼容性

- **SQL/2003** 核心特性。

### 示例

以下语句返回值 3749146.740。

```
SELECT SUM( Salary )  
FROM Employees;
```

## SUSER\_ID 函数 [System]

返回指定用户名的数字用户 ID。

### 语法

```
SUSER_ID( [ user-name ] )
```

### 参数

- **user-name** 您在搜索的用户 ID 的用户名。

### 返回值

INT

### 注释

如果不指定 *user-name*，则返回当前用户的 ID。

### 另请参见

- [“SUSER\\_NAME 函数 \[System\]” 一节 《SQL Anywhere 服务器 - SQL 参考》](#)
- [“USER\\_ID 函数 \[System\]” 一节 《SQL Anywhere 服务器 - SQL 参考》](#)

### 标准和兼容性

- **SQL/2003** Transact-SQL 扩充。

### 示例

以下语句返回 GROUPO 用户的 ID。

```
SELECT SUSER_ID( 'GROUPO' );
```

## SUSER\_NAME 函数 [System]

返回指定用户 ID 的用户名。

### 语法

```
SUSER_NAME([ user-id ])
```

### 参数

- **user-id** 您在搜索的用户的用户 ID。

### 返回值

LONG VARCHAR

### 注释

如果不指定 *user-id*，则返回当前用户的用户名。

### 另请参见

- “SUSER\_ID 函数 [System]” 一节 《SQL Anywhere 服务器 - SQL 参考》
- “USER\_NAME 函数 [System]” 一节 《SQL Anywhere 服务器 - SQL 参考》

### 标准和兼容性

- **SQL/2003** Transact-SQL 扩充。

### 示例

以下语句返回 ID 为 101 的用户的用户名。

```
SELECT SUSER_NAME( 101 );
```

## SYNC\_PROFILE\_OPTION\_VALUE 函数 [系统]

返回与给定选项名对应的选项的值。

### 语法

```
SYNC_PROFILE_OPTION_VALUE(profile_name, option_name)
```

### 参数

- **profile\_name** 要检查的同步配置文件的名称。
- **option\_name** 要为其检索对应值的选项的名称。

### 返回值

返回与给定选项名对应的选项的值。

### 注释

带有句点的选项名将从子列表中检索值，句点之前是给定的基本选项名，句点之后是给定的子列表选项名。

### 另请参见

- [“ML\\_GET\\_SERVER\\_NOTIFICATION \[系统\]”一节第 379 页](#)

### 标准和兼容性

- **SQL/2003** 服务商扩充。

### 示例

请考虑以下配置文件：

```
MobilinkUid=joe;Stream=tcip{host=sybase;port=1234};Ping=1
```

- **MobilinkUid** joe
- **stream** tcip{host=sybase;port=1234}
- **Stream.host** sybase
- **Stream.port** 1234
- **Ping** 1

## TAN 函数 [Numeric]

返回一个数字的正切值。

### 语法

**TAN**( *numeric-expression* )

### 参数

- **numeric-expression** 角度（以弧度为单位）。

### 返回值

DOUBLE

**注释**

ATAN 和 TAN 函数互为逆运算。

此函数将其参数转换为 DOUBLE，以双精度浮点执行计算，然后返回 DOUBLE 值作为结果。

**另请参见**

- [“COS 函数 \[Numeric\]”一节第 344 页](#)
- [“SIN 函数 \[Numeric\]”一节第 399 页](#)

**标准和兼容性**

- **SQL/2003** 服务商扩充。

**示例**

以下语句返回 0.52 的正切值。

```
SELECT TAN( 0.52 );
```

## TODAY 函数 [Date and time]

返回当前日期。

**语法**

```
TODAY( * )
```

**返回值**

DATE

**注释**

使用此语法代替以前的 CURRENT DATE 函数。

**标准和兼容性**

- **SQL/2003** 服务商扩充。

**示例**

以下语句根据系统时钟返回当前日期。

```
SELECT TODAY( * );  
SELECT CURRENT DATE;
```

## TRIM 函数 [String]

删除字符串中的前导和尾随空白。

**语法**

```
TRIM( string-expression )
```

**参数**

- **string-expression** 要剪裁的字符串。

**返回值**

VARCHAR  
 NVARCHAR  
 LONG VARCHAR  
 LONG NVARCHAR

**另请参见**

- [“LTRIM 函数 \[String\]”一节第 375 页](#)
- [“RTRIM 函数 \[String\]”一节第 394 页](#)
- [“UltraLite 字符串函数”一节第 327 页](#)

**标准和兼容性**

- **SQL/2003** TRIM 函数是 SQL/2003 的一个核心特性。

SQL Anywhere 不支持附加参数 *trim specification* 和 *trim character*（SQL/2003 中定义了这些参数）。TRIM 的 SQL Anywhere 实现对应于 BOTH 的 TRIM 说明。

针对其它由 SQL/2003 标准定义的 TRIM 说明（LEADING 和 TRAILING），SQL Anywhere 分别提供了 LTRIM 和 RTRIM 函数。

**示例**

以下语句返回没有前导和尾随空白的值 chocolate。

```
SELECT TRIM( '  chocolate  ' );
```

## TRUNCNUM 函数 [Numeric]

在小数点后的指定位数截断数字。

**语法**

```
{ TRUNCNUM | "TRUNCATE" }( numeric-expression, integer-expression )
```

**参数**

- **numeric-expression** 要截断的数字。
- **integer-expression** 正整数指定小数点右边舍入到的有效位数。负数表达式指定小数点左边舍入到的有效位数。

**返回值**

NUMERIC

**注释**

如果任何参数为 NULL，则结果为 NULL。

**另请参见**

- [“ROUND 函数 \[Numeric\]” 一节第 394 页](#)

**标准和兼容性**

- **SQL/2003** 服务商扩充。

**示例**

以下语句返回值 600。

```
SELECT TRUNCNUM( 655, -2 );
```

以下语句返回值 655.340。

```
SELECT TRUNCNUM( 655.348, 2 );
```

## TSEQUAL 函数 [System] (不再支持)

比较两个时间戳的值并返回它们是否相同。

**语法**

```
TSEQUAL ( timestamp1, timestamp2 )
```

**参数**

- **timestamp1** 时间戳表达式。
- **timestamp2** 时间戳表达式。

**返回值**

BIT

**注释**

TSEQUAL 函数只能用于 WHERE 子句中，并且大部分情况下作为 UPDATE 语句的一部分来使用。

如果 *timestamp1* 与 *timestamp2* 相等，则说明某行在读取后曾被更改。如果该行被更改了，则其时间戳会被修改过并且 TSEQUAL 函数会返回 FALSE。如果 TSEQUAL 函数返回 FALSE，应用程序确定没有更新过的行并假定该行被其它用户修改过。更新的行将重新读取。

可以使用 TSEQUAL 函数确定某行在读取后是否被更改过。

**另请参见**

- [“时间戳列的数据类型” 一节 《SQL Anywhere 服务器 - SQL 的用法》](#)
- [“TIMESTAMP 特殊值” 一节 《SQL Anywhere 服务器 - SQL 参考》](#)
- [“特殊的 Transact-SQL 时间戳列和数据类型” 一节 《SQL Anywhere 服务器 - SQL 的用法》](#)

## 标准和兼容性

- **SQL/2003** 服务商扩充。

## 示例

假设创建一个 `TIMESTAMP` 列 `Products.LastUpdated` 来存储该行上次更新的时间戳。以下的 `UPDATE` 语句使用 `TSEQUAL` 函数来确定是否更新该行。如果 `LastUpdated` 的值为 `'2010/12/25 11:08:34.173226'`，将更新该行。

```
UPDATE Products
SET Color = 'Yellow'
WHERE ID = '300'
AND TSEQUAL( LastUpdated, '2010/12/25 11:08:34.173226' );
```

## UCASE 函数 [String]

将字符串中的所有字符转换成大写形式。此函数与 `UPPER` 函数相同。

## 语法

**UCASE**( *string-expression* )

## 参数

- **string-expression** 要转换成大写形式的字符串。

## 返回值

VARCHAR

NVARCHAR

LONG VARCHAR

LONG NVARCHAR

## 另请参见

- [“UPPER 函数 \[String\]” 一节第 414 页](#)
- [“LCASE 函数 \[String\]” 一节第 368 页](#)
- [“UltraLite 字符串函数” 一节第 327 页](#)

## 标准和兼容性

- **SQL/2003** 服务商扩充。

## 示例

下面的语句返回值 `CHOCOLATE`。

```
SELECT UCASE( 'ChocoLate' );
```

## UPPER 函数 [String]

将字符串中的所有字符转换成大写形式。此函数与 UCASE 函数完全相同。

### 语法

**UPPER**( *string-expression* )

### 参数

- **string-expression** 要转换成大写形式的字符串。

### 返回值

VARCHAR  
NVARCHAR  
LONG VARCHAR  
LONG NVARCHAR

### 注释

UCASE 函数类似于 UPPER 函数。

### 另请参见

- “UCASE 函数 [String]” 一节第 413 页
- “LCASE 函数 [String]” 一节第 368 页
- “LOWER 函数 [String]” 一节第 374 页
- “UltraLite 字符串函数” 一节第 327 页

### 标准和兼容性

- **SQL/2003** 服务商扩充。

### 示例

下面的语句返回值 CHOCOLATE。

```
SELECT UPPER( 'ChocoLate' );
```

## USER\_ID 函数 [System]

返回指定用户名的数字用户 ID。

### 语法

**USER\_ID**( [ *user-name* ] )

### 参数

- **user-name** 您在搜索的用户 ID 的用户名。



## 返回值

INT

## 注释

如果不指定 *user-name*, 则返回当前用户的 ID。

## 另请参见

- “USER\_NAME 函数 [System]” 一节 《SQL Anywhere 服务器 - SQL 参考》
- “SUSER\_ID 函数 [System]” 一节 《SQL Anywhere 服务器 - SQL 参考》

## 标准和兼容性

- **SQL/2003** Transact-SQL 扩充。

## 示例

以下语句返回 GROUPO 用户 ID。

```
SELECT USER_ID( 'GROUPO' );
```

# USER\_NAME 函数 [System]

返回指定用户 ID 的用户名。

## 语法

```
USER_NAME([ user-id ])
```

## 参数

- **user-id** 您在搜索的用户的用户 ID。

## 返回值

LONG VARCHAR

## 注释

如果不指定 *user-id*, 则返回当前用户的用户名。

## 另请参见

- “USER\_ID 函数 [System]” 一节 《SQL Anywhere 服务器 - SQL 参考》
- “SUSER\_NAME 函数 [System]” 一节 《SQL Anywhere 服务器 - SQL 参考》

## 标准和兼容性

- **SQL/2003** 服务商扩充。

## 示例

以下语句返回 ID 为 101 的用户的用户名。

```
SELECT USER_NAME( 101 );
```

## UUIDTOSTR 函数 [String]

将唯一标识符值（UUID，又称 GUID）转换为字符串值。

### 较新的数据库中不需要

在使用 9.0.2 以前的版本创建的数据库中，需要使用 STRTOUUID 和 UUIDTOSTR 函数在 UUID 值的二进制与字符串表示之间进行转换。

在使用 9.0.2 或更高版本创建的数据库中，UNIQUEIDENTIFIER 数据类型已更改为本地数据类型，。在这些版本中不需要使用 STRTOUUID 和 UUIDTOSTR 函数。

有关详细信息，请参见“[UltraLite 中的数据类型](#)”一节第 290 页。

### 语法

```
UUIDTOSTR( uuid-expression )
```

### 参数

- **uuid-expression** 唯一标识符值。

### 返回值

VARCHAR

### 注释

将唯一标识符转换为 `xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx`（其中 x 是十六进制数字）格式的字符串值。如果此二进制值不是有效的唯一标识符，则返回 NULL。

如果您想查看 UUID 值，则可使用此函数。

### 另请参见

- “[NEWID 函数 \[Miscellaneous\]](#)”一节第 383 页
- “[STRTOUUID 函数 \[String\]](#)”一节第 404 页
- “[UltraLite 字符串函数](#)”一节第 327 页

### 标准和兼容性

- **SQL/2003** 服务商扩充。

### 示例

以下语句创建一个包含两列的 `mytab` 表。列 `pk` 的数据类型是唯一标识符，列 `c1` 的数据类型是整数。随后该语句在列 `c1` 中插入两行（值分别为 1 和 2）。

```
CREATE TABLE mytab(  
    pk UNIQUEIDENTIFIER PRIMARY KEY DEFAULT NEWID(),  
    c1 INT );  
INSERT INTO mytab( c1 ) values ( 1 );  
INSERT INTO mytab( c1 ) values ( 2 );
```

执行以下 SELECT 语句后，将返回新创建的表中的所有数据。

```
SELECT * FROM mytab;
```

您将看到一个包含两列、两行的表。为列 `pk` 显示的值将是二进制值。

若要将唯一标识符值转换为可读的格式，请执行以下的命令：

```
SELECT UUIDTOSTR(pk), c1 FROM mytab;
```

对于使用 9.0.2 或更高版本创建的数据库，不需要 `UUIDTOSTR` 函数。

## WEEKS 函数 [Date and time]

返回两个日期之间的周数。

### 语法 1

```
WEEKS( [ datetime-expression, ] datetime-expression )
```

### 语法 2

```
WEEKS( datetime-expression, integer-expression )
```

### 参数

- ***datetime-expression*** 日期和时间。
- ***integer-expression*** 要添加到 *datetime-expression* 中的周数。如果 *integer-expression* 是负数，则从日期时间值中减去相应的周数。如果提供 *integer-expression*，则必须将 *datetime-expression* 显式地转换为 DATETIME。

### 返回值

语法 1 返回 INTEGER。

语法 2 返回 TIMESTAMP。

### 注释

如果给定一个单独的日期（语法 1），`WEEKS` 函数将返回自 0000-02-29 后的周数。

如果给定两个日期（语法 1），`WEEKS` 函数将返回两者之间的周数。`WEEKS` 函数与 `DATEDIFF` 函数类似，但它们用于计算两个不同日期之间周数的方法不同，而且可以返回不同的结果。`WEEKS` 的返回值的确定方式是，将两个日期期间的天数除以七，然后向下舍入；然而，`DATEDIFF` 使用的是周边界数。这会导致返回的值不同。例如，如果第一个日期为星期五，第二个日期为下一个星期一，则 `WEEKS` 函数会返回差值 0，`DATEDIFF` 函数会返回差值 1。当两种方法相对彼此而言均无优势时，则在 `WEEKS` 和 `DATEDIFF` 之间进行选择时应考虑到差值。

有关 `DATEDIFF` 函数的详细信息，请参见“[DATEDIFF 函数 \[Date and time\]](#)”一节第 349 页。

如果给定一个日期和一个整数（语法 2），`WEEKS` 函数会将该整周数与指定日期相加。此函数与 `DATEADD` 函数类似。

有关 `DATEADD` 函数的详细信息，请参见“[DATEADD 函数 \[Date and time\]](#)”一节第 348 页。

### 另请参见

有关转换数据类型的信息，请参见“[CAST 函数 \[Data type conversion\]](#)”一节第 337 页。

### 标准和兼容性

- **SQL/2003** 服务商扩充。

### 示例

以下语句返回值 8，表示 2008-09-13 10:07:12 比 2008-07-13 06:07:12 晚八周。

```
SELECT WEEKS( '2008-07-13 06:07:12',  
              '2008-09-13 10:07:12' );
```

以下语句返回值 104792，表示该日期比 0000-02-29 晚 104792 周。

```
SELECT WEEKS( '2008-07-13 06:07:12' );
```

以下语句返回时间戳 2008-06-16 21:05:07.0，表示该日期和时间比 2008-05-12 21:05:07 晚五周。

```
SELECT WEEKS( CAST( '2008-05-12 21:05:07'  
                   AS TIMESTAMP ), 5);
```

## YEAR 函数 [Date and time]

以时间戳值为参数，返回该时间戳所指定的年份。

### 语法

```
YEAR( datetime-expression )
```

### 参数

- **datetime-expression** 一个日期、时间或时间戳。

### 返回值

SMALLINT

### 注释

该值作为 SMALL INT 返回。

### 标准和兼容性

- **SQL/2003** 服务商扩充。

### 示例

以下示例返回值 2001。

```
SELECT YEAR( '2001-09-12' );
```

## YEARS 函数 [Date and time]

给定两个日期时，此函数返回二者之间的整数年数。建议改用 DATEDIFF 函数。请参见“DATEDIFF 函数 [Date and time]”一节第 349 页。

给定一个日期时，此函数返回年份。建议改用 DATEPART 函数。请参见“DATEPART 函数 [Date and time]”一节《SQL Anywhere 服务器 - SQL 参考》。

给定一个日期和一个整数时，此函数为指定日期添加整数年数。建议改用 DATEADD 函数。请参见“DATEADD 函数 [Date and time]”一节《SQL Anywhere 服务器 - SQL 参考》。

### 语法 1

```
YEARS( [ datetime-expression, ] datetime-expression )
```

### 语法 2

```
YEARS( datetime-expression, integer-expression )
```

### 参数

- **datetime-expression** 日期和时间。
- **integer-expression** 要添加到 *datetime-expression* 中的年数。如果 *integer-expression* 是负数，则从日期时间值中减去相应的年数。如果提供 *integer-expression*，则必须将 *datetime-expression* 显式地转换为日期时间数据类型。

有关转换数据类型的信息，请参见“CAST 函数 [Data type conversion]”一节第 337 页。

### 返回值

语法 1 返回 INTEGER。语法 2 返回 TIMESTAMP。

### 注释

YEARS 值是从两个日期之间年中第一天的数目计算出的。

### 标准和兼容性

- **SQL/2003** 服务商扩充。

### 示例

以下语句都返回 -4。

```
SELECT YEARS( '1998-07-13 06:07:12',
              '1994-03-13 08:07:13' );

SELECT DATEDIFF( year,
                 '1998-07-13 06:07:12',
                 '1994-03-13 08:07:13' );
```

以下语句都返回 1998。

```
SELECT YEARS( '1998-07-13 06:07:12' )
SELECT DATEPART( year, '1998-07-13 06:07:12' );
```

以下语句返回给定日期 300 年后的日期。

```
SELECT YEARS( CAST( '1998-07-13 06:07:12' AS TIMESTAMP ), 300 )  
SELECT DATEADD( year, 300, '1998-07-13 06:07:12' );
```

## YMD 函数 [Date and time]

返回对应于给定年、月、日的日期值。值是 -32768 到 32767 的小整数。

### 语法

```
YMD(  
integer-expression1,  
integer-expression2,  
integer-expression3 )
```

### 参数

- **integer-expression1** 年。
- **integer-expression2** 表示月份的数字。如果月份超出 1-12 的范围，则会调整年份。
- **integer-expression3** 天数。日可以是任意整数；日期可相应调整。

### 返回值

DATE

### 标准和兼容性

- **SQL/2003** 服务商扩充。

### 示例

以下语句返回值 1998-06-12。

```
SELECT YMD( 1998, 06, 12 );
```

如果值超出其正常范围，则会调整日期。例如，以下语句返回值 2000-03-01。

```
SELECT YMD( 1999, 15, 1 );
```

---

# UltraLite SQL 语句

## 目录

UltraLite 语句类别 .....	423
UltraLite ALTER DATABASE SCHEMA FROM FILE 语句 .....	424
UltraLite ALTER PUBLICATION 语句 .....	425
UltraLite ALTER SYNCHRONIZATION PROFILE 语句 .....	426
UltraLite ALTER TABLE 语句 .....	428
UltraLite CHECKPOINT 语句 .....	431
UltraLite COMMIT 语句 .....	432
UltraLite CREATE INDEX 语句 .....	433
UltraLite CREATE PUBLICATION 语句 .....	435
UltraLite CREATE SYNCHRONIZATION PROFILE 语句 .....	436
UltraLite CREATE TABLE 语句 .....	440
UltraLite DELETE 语句 .....	444
UltraLite DROP INDEX 语句 .....	445
UltraLite DROP PUBLICATION 语句 .....	446
UltraLite DROP SYNCHRONIZATION PROFILE 语句 .....	447
UltraLite DROP TABLE 语句 .....	448
UltraLite FROM 子句 .....	449
UltraLite INSERT 语句 .....	451
UltraLite LOAD TABLE 语句 .....	452
UltraLite ROLLBACK 语句 .....	456
UltraLite SELECT 语句 .....	457
UltraLite SET OPTION 语句 .....	459
UltraLite START SYNCHRONIZATION DELETE 语句 .....	460
UltraLite STOP SYNCHRONIZATION DELETE 语句 .....	461
UltraLite SYNCHRONIZE 语句 .....	462
UltraLite TRUNCATE TABLE 语句 .....	464
UltraLite UNION 语句 .....	466
UltraLite UPDATE 语句 .....	467

UltraLite SQL 支持 SQL 语句，它们是 SQL Anywhere 数据库所支持的语句的子集。

### 开始之前

- UltraLite 中的表不支持所有者的概念。为了便于使用现有的 SQL 和以编程方式生成的 SQL，UltraLite 仍然允许使用语法 *owner.table-name*。但是不会检查所有者，因为 UltraLite 不支持表的所有者。
- UltraLite SQL 语句文档遵循的语法约定与 SQL Anywhere 语句使用的语法约定相同。确保理解这些约定以及它们如何用于表示 SQL 语法。请参见“[语法约定](#)”一节《[SQL Anywhere 服务器 - SQL 参考](#)》。
- 使用 UltraLite SQL 创建事务。事务包括自上次执行 ROLLBACK 或 COMMIT 以来所做的所有更改（INSERT、UPDATE 和 DELETE）。请参见“[UltraLite 事务处理](#)”一节第 14 页。  
这些更改可通过执行 COMMIT 语句永久生效。ROLLBACK 语句可使更改被删除。请参见“[UltraLite COMMIT 语句](#)”一节第 432 页和“[UltraLite ROLLBACK 语句](#)”一节第 456 页。
- 要查找 Interactive SQL 使用的语句，请参见“[SQL 语句](#)”《[SQL Anywhere 服务器 - SQL 参考](#)》。Interactive SQL 使用的语句在语句名后面有 **[Interactive SQL]**。例如，“[CONFIGURE 语句 \[Interactive SQL\]](#)”一节《[SQL Anywhere 服务器 - SQL 参考](#)》。



## UltraLite 语句类别

SQL 语句由语句中的初始字组织和定义，且初始字几乎始终是动词。此面向操作的语法使得该语言的性质变为一组对数据库的命令语句（命令）。在 UltraLite 中，支持的 SQL 语句可分为以下几类：

- **数据检索语句** 也称作查询。这些语句可用于从表中选择数据表达式的行。数据检索可使用 SELECT 语句来实现。请参见“[UltraLite SELECT 语句](#)”一节第 457 页。
- **数据操作语句** 可用于更改数据库中的内容。数据操作可使用以下语句来实现：
  - “[UltraLite DELETE 语句](#)”一节第 444 页
  - “[UltraLite INSERT 语句](#)”一节第 451 页
  - “[UltraLite UPDATE 语句](#)”一节第 467 页
- **数据定义语句** 可用于定义数据库的结构或模式。该模式可用以下语句进行更改：
  - “[UltraLite ALTER DATABASE SCHEMA FROM FILE 语句](#)”一节第 424 页
  - “[UltraLite CREATE INDEX 语句](#)”一节第 433 页
  - “[UltraLite CREATE TABLE 语句](#)”一节第 440 页
  - “[UltraLite DROP INDEX 语句](#)”一节第 445 页
  - “[UltraLite DROP TABLE 语句](#)”一节第 448 页
  - “[UltraLite ALTER TABLE 语句](#)”一节第 428 页
  - “[UltraLite TRUNCATE TABLE 语句](#)”一节第 464 页
- **事务控制语句** 可用于在 UltraLite 应用程序中控制事务。事务控制可使用以下语句来实现：
  - “[UltraLite CHECKPOINT 语句](#)”一节第 431 页
  - “[UltraLite COMMIT 语句](#)”一节第 432 页
  - “[UltraLite ROLLBACK 语句](#)”一节第 456 页
- **同步管理** 可用于暂时控制与 MobiLink 服务器的同步。同步管理可使用以下语句来实现：
  - “[UltraLite START SYNCHRONIZATION DELETE 语句](#)”一节第 460 页
  - “[UltraLite STOP SYNCHRONIZATION DELETE 语句](#)”一节第 461 页
  - “[UltraLite CREATE PUBLICATION 语句](#)”一节第 435 页
  - “[UltraLite ALTER PUBLICATION 语句](#)”一节第 425 页
  - “[UltraLite DROP PUBLICATION 语句](#)”一节第 446 页

### 另请参见

- “[UltraLite 中的表达式](#)”一节第 303 页
- “[UltraLite 中的运算符](#)”一节第 316 页

# UltraLite ALTER DATABASE SCHEMA FROM FILE 语句

此语句用于在 SQL 脚本中修改现有 UltraLite 数据库的模式定义。

## 语法

```
ALTER DATABASE SCHEMA FROM FILE filename
```

## 参数

**filename** 定义用于升级现有 UltraLite 数据库模式的 SQL 脚本的名称和路径。

## 注释

使用 `ulinit` 或 `ulunload` 抽取您的脚本所需的 DDL 语句。通过使用这些实用程序，可确保 DDL 语句在语法上是正确的。使用 `ulinit` (`-l logfile` 选项) 或 `ulunload` (使用 `-n -s output-file` 选项)。请参见“UltraLite 初始化数据库实用程序 (`ulinit`)”一节第 249 页和“UltraLite 卸载旧数据库实用程序 (`ulunload`)”一节第 260 页。

执行本语句前备份数据库。

SQL 脚本文件的字符集必须匹配您想升级的数据库的字符集。

执行本语句时要确保未重置您的设备。如果在模式升级过程中重置了设备，则 UltraLite 数据库将变得不可用。

将删除不适合模式的所有行（例如，如果添加了唯一性约束并且多行包含相同的值，则仅保留一行而删除其它的所有行）。这种情况下会生成

`SQL_ROW_DROPPED_DURING_SCHEMA_UPGRADE` 警告。可使用此警告检测错误，并从备份版本中恢复数据库。

## 另请参见

- “部署 UltraLite 模式升级”一节第 58 页

## 示例

以下语句在 SQL 脚本中修改数据库的模式，`MySchema.sql`：

```
ALTER DATABASE SCHEMA FROM FILE 'MySchema.sql';
```

## UltraLite ALTER PUBLICATION 语句

此语句用于变更发布。发布可以标识远程数据库中的同步数据。

### 语法

```
ALTER PUBLICATION publication-name alterpub-clause
```

*alterpub-clause* :

```
ADD TABLE table-name [ WHERE search-condition ]  
| ALTER TABLE table-name [ WHERE search-condition ]  
| { DROP | DELETE } TABLE table-name  
| RENAME publication-name
```

### 副作用

自动提交。

### 另请参见

- “UltraLite 中的搜索条件” 一节第 309 页
- “设计 UltraLite 中的同步” 一节第 123 页
- “UltraLite CREATE PUBLICATION 语句” 一节第 435 页
- “UltraLite DROP PUBLICATION 语句” 一节第 446 页
- “UltraLite START SYNCHRONIZATION DELETE 语句” 一节第 460 页
- “UltraLite STOP SYNCHRONIZATION DELETE 语句” 一节第 461 页

### 示例

以下 ALTER PUBLICATION 语句将 Customers 表添加到 pub\_contact 发布中。

```
ALTER PUBLICATION pub_contact  
ADD TABLE Customers;
```

## UltraLite ALTER SYNCHRONIZATION PROFILE 语句

此语句用于变更 UltraLite 同步配置文件。同步配置文件定义 UltraLite 数据库如何与 MobiLink 服务器同步。

### 语法

```
ALTER SYNCHRONIZATION PROFILE sync-profile-name  
{ REPLACE | MERGE } sync-option [; ... ]
```

*sync-option* :

*sync-option-name* = *sync-option-value*

*sync-option-name* : *string*

*sync-option-value* : *string*

### 参数

- **sync-profile-name** 同步配置文件的名称。
- **REPLACE 子句** 此子句用于删除当前为配置文件定义的选项，并添加指定的替代选项。
- **MERGE 子句** 此子句用于更改同步配置文件的现有选项或向其中添加新选项。
- **sync-option** 一个或多个同步选项值对的字符串用分号分隔。例如，'option1=value1;option2=value2'。
- **sync-option-name** 同步配置文件选项的名称。
- **sync-option-value** 同步配置文件选项的值。

### 注释

有关 UltraLite 所支持的同步配置文件选项的列表，请参见“[UltraLite CREATE SYNCHRONIZATION PROFILE 语句](#)”一节第 436 页。

### 副作用

无。

### 另请参见

- [“UltraLite DROP SYNCHRONIZATION PROFILE 语句”一节第 447 页](#)
- [“UltraLite SYNCHRONIZE 语句”一节第 462 页](#)

### 示例

下面是 ALTER SYNCHRONIZATION PROFILE...REPLACE 语句的示例：

```
CREATE SYNCHRONIZATION PROFILE myProfile1;  
ALTER SYNCHRONIZATION PROFILE myProfile1  
  REPLACE 'publication=p1;uploadonly=on';
```

下面是 ALTER SYNCHRONIZATION PROFILE...MERGE 语句的示例。

```
CREATE SYNCHRONIZATION PROFILE myProfile2 'publication=p1;verbosity=high';  
ALTER SYNCHRONIZATION PROFILE myProfile2  
    MERGE 'publication=p2;uploadonly=on';
```

## UltraLite ALTER TABLE 语句

此语句用于修改表定义。

### 语法

```
ALTER TABLE table-name {  
  add-clause  
  | modify-clause  
  | drop-clause  
  | rename-clause  
}
```

```
add-clause :  
ADD { column-definition | table-constraint }
```

```
modify-clause :  
ALTER column-definition
```

```
drop-clause :  
DROP { column-name | CONSTRAINT constraint-name }
```

```
rename-clause :  
RENAME {  
  new-table-name  
  | [ old-column-name TO ] new-column-name  
  | CONSTRAINT old-constraint-name TO new-constraint-name }
```

```
column-definition :  
column-name data-type  
  [ [ NOT ] NULL ]  
  [ DEFAULT column-default ]  
  [ UNIQUE ]
```

```
column-default :  
GLOBAL AUTOINCREMENT [ ( number ) ]  
| AUTOINCREMENT  
| CURRENT DATE  
| CURRENT TIME  
| CURRENT TIMESTAMP  
| NULL  
| NEWID( )  
| constant-value
```

```
table-constraint :  
[ CONSTRAINT constraint-name ]  
{ fkey-constraint | unique-key-constraint }  
[ WITH MAX HASH SIZE integer ]
```

```
fkey-constraint :  
[ NOT NULL ] FOREIGN KEY [ role-name ] ( ordered-column-list )  
  REFERENCES table-name ( column-name, ... )  
  [ CHECK ON COMMIT ]
```

*unique-key-constraint* :  
**UNIQUE** ( *ordered-column-list* )

*ordered-column-list* :  
 ( *column-name* [ **ASC** | **DESC** ], ... )

## 参数

**add-clause** 向表中添加新列或表约束：

- **ADD *column-definition* 子句** 向表中添加新列。如果列有缺省值，新列中的所有行将用该缺省值来填充。有关对该子句的关键字和次级子句的说明，请参见“UltraLite CREATE TABLE 语句”一节第 440 页。
- **ADD *table-constraint* 子句** 向表中添加约束。可选约束名用于以后单独修改或删除约束，而不必修改整个表的约束。有关对该子句的关键字和次级子句的说明，请参见“UltraLite CREATE TABLE 语句”一节第 440 页。

**注意**  
 不能在 UltraLite 中添加主键。

**modify-clause** 更改单个列的定义。请注意，当语句是 ALTER 语句的一部分时，无法在 *column-definition* 中使用主键。有关 *column-definition* 的完整说明，请参见“UltraLite CREATE TABLE 语句”一节第 440 页。

**drop-clause** 删除列或表约束：

- **DROP *column-name*** 从表中删除列。如果列包含在任何索引、唯一约束、外键或主键中，则必须删除此对象后 UltraLite 才能删除该列。
- **DROP CONSTRAINT *table-constraint*** 从表的定义中删除命名约束。有关 *table-constraint* 的完整说明，请参见“UltraLite CREATE TABLE 语句”一节第 440 页。

**注意**  
 不能在 UltraLite 中删除主键。

**rename-clause** 更改表、列或约束的名称：

- **RENAME *new-table-name*** 将表名更改为 *new-table-name*。请注意，必须修改所有使用旧表名的应用程序。被自动分配旧表名的外键的名称不更改。
- **RENAME *old-column-name* TO *new-column-name*** 将列名更改为 *new-column-name*。请注意，需要修改所有使用旧列名的应用程序。
- **RENAME *old-constraint-name* TO *new-constraint-name*** 将约束名更改为 *new-constraint-name*。请注意，需要修改所有使用旧约束名的应用程序。

**注意**  
 不能在 UltraLite 中重命名主键。

**column-constraint** 为了确保数据库中数据的完整性，列约束将对列能够保存的值进行限制。列约束只能是 UNIQUE。

**UNIQUE** 标识唯一标识表中各行的一列或多列。表中任何两行的值在所有指定的列中不能相同。表可以有多个唯一约束。

## 注释

在一个 ALTER TABLE 语句中，只能添加、修改或删除一个 *table-constraint* 或 *column-constraint*。

角色名是外键的名称。*role-name* 的主要作用是区分同一表的两个外键。或者，可使用 CONSTRAINT *constraint-name* 命名外键。但是，请勿同时使用两种方法来命名外键。

不能通过 MODIFY 修改表约束或列约束。要更改约束，必须通过 DELETE 删除旧约束并通过 ADD 添加新约束。

名称以 **nosync** 结尾的表只能重命名为也以 **nosync** 结尾的表名。请参见“UltraLite 中的 nosync 表”一节第 124 页。

如果影响表的语句已经由其它请求或查询引用，则 ALTER TABLE 语句将无法执行。同样，当表被变更时，UltraLite 不会处理引用表的请求。此外，当数据库包含活动查询或未提交的事务时，您也无法执行 ALTER TABLE 语句。

对于 UltraLite.NET 用户：除非也为所有数据对象（例如 ULDataReader）调用了 Dispose 方法，否则您将无法执行此语句。请参见“Dispose 方法”一节《UltraLite - .NET 编程》。

如果模式更改同时开始，则语句将不会被释放。请参见“使用 DDL 语句更改模式”一节第 9 页。

## 另请参见

- “UltraLite CREATE TABLE 语句”一节第 440 页
- “UltraLite DROP TABLE 语句”一节第 448 页
- “UltraLite 中的数据类型”一节第 290 页
- “变更表”一节《SQL Anywhere 服务器 - SQL 的用法》
- “使用表和列约束”一节《SQL Anywhere 服务器 - SQL 的用法》
- “替换自动增量列的分区大小”一节第 121 页
- “确定最近指派的 GLOBAL AUTOINCREMENT 值”一节第 120 页

## 示例

以下语句从名为 MyEmployees 的虚构表中删除 Street 列。

```
ALTER TABLE MyEmployees
DROP Street;
```

以下示例将 MyCustomers 虚构表的 Street 列更改为可容纳约 50 个字符。

```
ALTER TABLE MyCustomers
MODIFY Street CHAR(50);
```



## UltraLite CHECKPOINT 语句

此语句用于对数据库执行检查点操作。

### 语法

**CHECKPOINT**

### 注释

可使用 CHECKPOINT 语句作为提交刷新的触发器。事务刷新将未提交的事务写入存储。

如果使用嵌入式 SQL API，则也可使用 ULCheckpoint 方法。如果编写 C++ 组件应用程序，则也可在连接对象中使用 Checkpoint 方法。所有其它 API 都必须使用此语句。

### 副作用

虽然此语句将所有待执行的已提交事务刷新到存储，但它不提交或刷新当前事务。

### 另请参见

- [“刷新单个或分组的事务”一节第 110 页](#)
- [“UltraLite COMMIT 语句”一节第 432 页](#)
- [“UltraLite COMMIT\\_FLUSH 连接参数”一节第 210 页](#)
- UltraLite 嵌入式 SQL: [“ULCheckpoint 函数”一节《UltraLite - C 及 C++ 编程》](#)
- UltraLite C++: [“Checkpoint 函数”一节《UltraLite - C 及 C++ 编程》](#)

### 示例

以下语句执行数据库的检查点。

```
CHECKPOINT;
```

## UltraLite COMMIT 语句

使用此语句使对数据库进行的更改成为永久更改。

### 语法

```
COMMIT [ WORK ]
```

### 注释

使用 UltraLite SQL 创建事务。事务包括自上次执行 ROLLBACK 或 COMMIT 以来所做的所有更改（INSERT、UPDATE 和 DELETE）。COMMIT 语句结束当前事务，并使在该事务期间所做的所有更改在数据库中永久生效。

使用 ALTER、CREATE 和 DROP 语句对数据库对象的更改是自动提交的。

### 另请参见

- [“UltraLite CHECKPOINT 语句”一节第 431 页](#)
- [“UltraLite ROLLBACK 语句”一节第 456 页](#)

### 示例

以下语句使当前事务中的更改在数据库中永久生效：

```
COMMIT;
```

## UltraLite CREATE INDEX 语句

此语句用于在指定表中创建索引。

### 语法

```
CREATE [ UNIQUE ] INDEX [ index-name ]  
ON table-name ( ordered-column-list )  
[ WITH MAX HASH SIZE integer ]
```

```
ordered-column-list :  
( column-name [ ASC | DESC ], ... )
```

### 参数

**UNIQUE** UNIQUE 参数确保表中不会有两行的值在所有索引列中相同。每个索引键都必须是唯一的，或者至少在一列中包含 NULL。

表的唯一约束和唯一索引是有区别的。唯一索引中的列可以为 NULL，而唯一约束的列不能为 NULL。同时，外键可以引用主键或者有唯一约束的列。但是，外键不能引用唯一索引。

**ordered-column-list** 列的有序列表。索引中列的值可按升序或降序排列。

**WITH MAX HASH SIZE** 为此索引设置散列大小（按字节计）。该值替换对数据库有效的缺省 MaxHashSize 属性。要学习缺省大小，请参见“[访问 UltraLite 数据库属性](#)”一节第 199 页。

### 注释

UltraLite 自动创建主键和唯一约束的索引。

索引为 UltraLite 查找特定行提供了快捷的方法，因此提高了查询性能。相反，由于必须维护索引，所以索引可能降低同步以及 INSERT、DELETE 和 UPDATE 语句的速度。

此索引自动用于改进向数据库发出的查询的性能，以及通过 ORDER BY 子句对查询进行排序。索引一旦创建后就再也不能在 SQL 语句中再次引用，除非使用 DROP INDEX 将它删除。

索引会占用数据库中的空间。此外，维护索引需要的额外操作会影响数据修改操作的性能。由于这些原因，应避免创建不能提高查询性能的索引。

当 CREATE INDEX 语句正在被处理时，UltraLite 不会处理引用该索引的请求或查询。此外，当数据库包含活动查询或未提交的事务时，您也无法执行 CREATE INDEX 语句。

UltraLite 也可使用执行计划来优化查询。请参见“[UltraLite 中的执行计划](#)”一节第 320 页。

对于 UltraLite.NET 用户：除非也为所有数据对象（例如 ULDataReader）调用了 Dispose 方法，否则您将无法执行此语句。请参见“[Dispose 方法](#)”一节《[UltraLite - .NET 编程](#)》。

如果模式更改同时开始，则语句将不会被释放。请参见“[使用 DDL 语句更改模式](#)”一节第 9 页。

### 副作用

- 自动提交。

### 另请参见

- [“UltraLite 性能与优化” 第 99 页](#)
- [“UltraLite DROP INDEX 语句” 一节第 445 页](#)
- [“使用 UltraLite 索引” 一节第 71 页](#)

### 示例

以下语句在 Employees 表上创建一个由两个列组成的索引。

```
CREATE INDEX employee_name_index  
ON Employees ( Surname, GivenName );
```

以下语句在 SalesOrderItems 表上为 ProductID 列创建索引。

```
CREATE INDEX item_prod  
ON SalesOrderItems ( ProductID );
```

## UltraLite CREATE PUBLICATION 语句

此语句用于创建发布。发布可以标识 UltraLite 远程数据库中的同步数据。

### 语法

```
CREATE PUBLICATION publication-name  
( TABLE table-name [ WHERE search-condition ], ... )
```

### 参数

**TABLE 子句** 此表用于在出版中包括 TABLE。对 TABLE 子句的数目没有限制

**WHERE 子句** 如果指定 WHERE 子句，则在同步期间将只考虑从关联表上载满足 *search-condition* 的行。请参见“[UltraLite 中的搜索条件](#)”一节第 309 页。

如果不指定 WHERE 子句，则会考虑上载表中自上一次同步以来在 UltraLite 中更改的每一行。

### 注释

一个发布可建立在单个同步操作期间同步的表，并决定哪些数据会被上载到 MobiLink 服务器。在下载会话期间，MobiLink 服务器可能会发回这些（且只发回这些）表的行，但是，下载的行不必满足表的 WHERE 子句。

只能发布完整的表。在 UltraLite 中，不能发布表的特定列。

### 副作用

- 自动提交。

### 另请参见

- “[UltraLite 客户端](#)”第 115 页
- “[UltraLite DROP PUBLICATION 语句](#)”一节第 446 页
- “[UltraLite ALTER PUBLICATION 语句](#)”一节第 425 页
- “[UltraLite 中的搜索条件](#)”一节第 309 页

### 示例

以下语句发布两个表中的所有列和行。

```
CREATE PUBLICATION pub_contact (  
    TABLE Contacts,  
    TABLE Customers  
);
```

以下语句仅发布 Customers 表（其中 State 列包含 MN）的行。

```
CREATE PUBLICATION pub_customer (  
    TABLE Customers  
    WHERE State = 'MN'  
);
```

## UltraLite CREATE SYNCHRONIZATION PROFILE 语句

此语句用于创建 UltraLite 同步配置文件。同步配置文件定义 UltraLite 数据库如何与 MobiLink 服务器同步。

### 语法

```
CREATE SYNCHRONIZATION PROFILE sync-profile-name sync-option [;...]
```

*sync-option* :

*sync-option-name* = *sync-option-value*

*sync-option-name* : string

*sync-option-value* : string

### 参数

- **sync-profile-name** 同步配置文件的名称。
- **sync-option** 一个或多个同步选项值对的字符串用分号分隔。例如, 'option1=value1;option2=value2'。
- **sync-option-name** 同步配置文件选项的名称。
- **sync-option-value** 同步配置文件选项的值。

### 注释

有两种方法可用于更改现有同步配置文件。第一种是使用 REPLACE 子句。这会将同步配置文件的内容替换为在新 sync-option 字符串中所包含的内容。这相当于删除同步配置文件然后创建一个具有相同名称但使用新字符串的同步配置文件。因此, 请注意, 同步配置文件无需包含完整的同步定义因为在同步时可以合并(或替换)参数。

第二种修改同步配置文件的方法是使用 MERGE 子句。使用此子句时, 仅更改在 MERGE 子句中指定的同步选项。要从同步配置文件中删除同步选项, sync-option 字符串应与 'option1=;' 类似(为了将该选项设置为空值)。

STREAM 同步配置文件选项与其它选项不同, 因为它的值包含 sub-list。例如: 'STREAM=TCPIP{host=192.168.1.1;port=1234}'。这种情况下 'host=192.168.1.1;port=1234' 是 sub-list。添加或删除 sub-list 值, 请使用在 STREAM sync-option-name 和 sub-option-name 之间的句点。例如, MERGE 'stream.port=5678;stream.host=;compression=zlib' 生成了如下同步配置文件: stream=TCPIP{port=5678;compression=zlib}。尝试将此流设置为新值将替换整个流的值。例如: MERGE 'stream=HTTPS' 生成了如下同步配置文件: stream=HTTPS{}。

下表列出了 UltraLite 所支持的同步配置文件选项。

同步配置文件选项	有效值	说明
AllowDownloadDupRows	布尔值	下载具有相同主键的多行时，本选项可防止错误的出现。此选项可用于同步不一致的数据，而不会导致同步失败。缺省值为 "no"。请参见“ <a href="#">Additional Parameters 同步参数</a> ”一节第 138 页。
AuthParms	字符串 (以逗号分隔)	指定验证参数列表发送到 MobiLink 服务器。可使用验证参数在 MobiLink 脚本中执行自定义验证。请参见“ <a href="#">Authentication Parameters 同步参数</a> ”一节第 140 页。
CheckpointStore	布尔值	添加数据库在同步过程中的额外检查点，以限制数据库在同步过程中的增长。请参见“ <a href="#">Additional Parameters 同步参数</a> ”一节第 138 页。
ContinueDownload	布尔值	重新启动以前执行失败的下载。继续下载时，仅能接收到在同步失败时所选择要下载的更改。缺省情况下，UltraLite 不继续下载。请参见“ <a href="#">恢复失败的下载</a> ”一节《 <a href="#">MobiLink - 服务器管理</a> 》。
DisableConcurrency	布尔值	在同步期间禁止从其它线程访问数据库。请参见“ <a href="#">Additional Parameters 同步参数</a> ”一节第 138 页。
DownloadOnly	布尔值	执行仅下载同步。请参见“ <a href="#">Download Only 同步参数</a> ”一节第 142 页。
KeepPartialDownload	布尔值	如果发生通信错误，此选项控制 UltraLite 是否保持部分下载。缺省情况下，UltraLite 不回退部分下载的更改。请参见“ <a href="#">Keep Partial Download 同步参数</a> ”一节第 144 页。
MobiLinkPwd	字符串	指定与用户名关联的现有 MobiLink 口令。请参见“ <a href="#">MobiLinkPwd (mp) 扩展选项</a> ”一节《 <a href="#">MobiLink - 客户端管理</a> 》。
MobiLinkUid	字符串	指定 MobiLink 用户名。请参见“-u 选项”一节《 <a href="#">MobiLink - 客户端管理</a> 》。请参见“-mn 选项”一节《 <a href="#">MobiLink - 客户端管理</a> 》。
NewMobiLinkPwd	字符串	为 MobiLink 用户提供新口令。在想要更改现有口令时，可使用此选项。请参见“-mn 选项”一节《 <a href="#">MobiLink - 客户端管理</a> 》。
Ping	布尔值	仅确认与服务器的通信，未执行同步。请参见“ <a href="#">Ping 同步参数</a> ”一节第 148 页。

同步配置文件选项	有效值	说明
Publications	字符串 (以逗号分隔)	指定要同步的发布。发布可确定远程数据库上包含在同步操作中的表。如果此参数为空白(缺省值),则所有表都是同步的。如果此参数为星号(*),则所有发布都是同步的。请参见“UltraLite 中的发布”一节第 124 页。
ScriptVersion	字符串	指定 MobiLink 脚本版本。脚本版本确定在同步期间 MobiLink 在统一数据库上运行的脚本。如果未指定脚本版本,则使用“缺省值”。请参见“ScriptVersion (sv) 扩展选项”一节《MobiLink - 客户端管理》。
SendColumnNames	字符串	同步时应指定将列名作为上载文件的一部分发送到 MobiLink 服务器。缺省情况下,不发送列名。请参见“Send Column Names 同步参数”一节第 151 页。
SendDownloadACK	布尔值	指定客户端应该向服务器发送一个下载确认。缺省情况下,MobiLink 服务器不提供下载确认。请参见“发送下载确认同步参数”一节第 152 页。
stream	字符串 (带有 sub-list)	指定 MobiLink 网络同步协议。请参见“Stream Type 同步参数”一节第 154 页。
TableOrder	字符串 (以逗号分隔)	指定上载中的表顺序。缺省情况下,UltraLite 选择基于外键关系的顺序。请参见“Additional Parameters 同步参数”一节第 138 页。
UploadOnly	字符串	指定同步仅包含上载,不进行任何下载。请参见“Upload Only 同步参数”一节第 157 页。

布尔值可被指定为 Yes/No、1/0、True/False、On/Off。在所有的布尔实例中,缺省值为“No”。对于所有其它值,完全未指定缺省值。

## 副作用

无。

## 另请参见

- “UltraLite ALTER SYNCHRONIZATION PROFILE 语句”一节第 426 页
- “UltraLite DROP SYNCHRONIZATION PROFILE 语句”一节第 447 页
- “UltraLite SYNCHRONIZE 语句”一节第 462 页

## 示例

以下创建的同步配置文件名为 Test1。

```
CREATE SYNCHRONIZATION PROFILE Test1
'MobiLinkUid=mary;Stream=TCPIP{host=192.168.1.1;port=1234}'
```



以下示例说明了在使用不同的选项执行 ALTER SYNCHRONIZATION PROFILE 命令序列后所发生的变化。

假设 myProfile1='MobiLinkUID=mary;ScriptVersion=default'。

执行 ALTER SYNCHRONIZATION PROFILE myProfile1 REPLACE 'MobiLinkPwd=sql;ScriptVersion=1' 后, myProfile1 为 'MobiLinkPwd=sql;ScriptVersion=1'。

执行 ALTER SYNCHRONIZATION PROFILE myProfile1 MERGE 'MobiLinkUID=mary;STREAM=tcpip' 后, myProfile1 为 'MobiLinkPwd=sql;ScriptVersion=1;MobiLinkUID=mary;STREAM=tcpip'。

执行 ALTER SYNCHRONIZATION PROFILE myProfile1 MERGE 'MobiLinkUID=;STREAM.host=192.168.1.1;STREAM.port=1234;ScriptVersion=;' 后, myProfile1 为 'MobiLinkPwd=sql;STREAM=tcpip{192.168.1.1;port=1234}'。

执行 ALTER SYNCHRONIZATION PROFILE myProfile1 MERGE 'MobiLinkPwd=;Ping=yes;STREAM =HTTP' 后, myProfile1 为 'Ping=yes;STREAM=HTTP'。

执行 ALTER SYNCHRONIZATION PROFILE myProfile1 MERGE 'STREAM=HTTP{host=192.168.1.1}' 后, myProfile1 为 'Ping=yes;STREAM=HTTP{host=192.168.1.1}'。

## UltraLite CREATE TABLE 语句

此语句用于创建表。

### 语法

```
CREATE TABLE table-name (  
  { column-definition | table-constraint }, ...  
)
```

```
column-definition :  
column-name data-type  
[ [ NOT ] NULL ]  
[ DEFAULT column-default ]  
[ column-constraint ]
```

```
column-default :  
GLOBAL AUTOINCREMENT [ ( number ) ]  
| AUTOINCREMENT  
| CURRENT DATE  
| CURRENT TIME  
| CURRENT TIMESTAMP  
| NULL  
| NEWID( )  
| constant-value
```

```
column-constraint :  
PRIMARY KEY  
| UNIQUE
```

```
table-constraint :  
{ [ CONSTRAINT constraint-name ]  
  pkey-constraint  
  | fkey-constraint  
  | unique-key-constraint }  
[ WITH MAX HASH SIZE integer ]
```

```
pkey-constraint :  
PRIMARY KEY [ ordered-column-list ]
```

```
fkey-constraint :  
[ NOT NULL ] FOREIGN KEY [ role-name ] ( ordered-column-list )  
  REFERENCES table-name ( column-name, ... )  
  [ CHECK ON COMMIT ]
```

```
unique-key-constraint :  
UNIQUE ( ordered-column-list )
```

```
ordered-column-list :  
( column-name [ ASC | DESC ], ... )
```

### 参数

**column-definition** 定义表中的列。此子句的可用参数包括：

- **column-name** 列名是一个标识符。同一表中的两列不能同名。请参见“[UltraLite 中的标识符](#)”一节第 281 页。
  - **data-type** 列的数据类型。请参见“[UltraLite 中的数据类型](#)”一节第 290 页。
  - **[ NOT ] NULL** 如果指定 NOT NULL，或者列具有 PRIMARY KEY 或 UNIQUE 约束，则该列在任何行中都不能含有 NULL。否则，允许包含 NULL。
  - **column-default** 设置列的缺省值。如果指定了 DEFAULT 值，则它将用作未指定列值的任何 INSERT 语句中的列值。如果未指定任何 DEFAULT 值，则它等效于 DEFAULT NULL。缺省值选项包括：
    - **AUTOINCREMENT** 使用 AUTOINCREMENT 时，列必须是整型数据类型之一或是精确的数字类型。插入到表中时，如果没有指定 AUTOINCREMENT 列的值，则生成一个比列中的任何其它值都大的唯一值。如果 INSERT 指定的列值大于列的当前最大值，则该值将用作后续插入的起点值。
- 提示**  
在 UltraLite 中，创建表时自动增量值不能设置为 0。如果该列使用有符号数据类型，AUTOINCREMENT 将生成负值。因此，应将 AUTOINCREMENT 列声明为无符号整数，以防止使用负值。
- **GLOBAL AUTOINCREMENT** 与 AUTOINCREMENT 类似，只不过要对域进行分区。每个分区都包含相同数目的值。为每个数据库副本指定一个唯一全局数据库标识号。UltraLite 只从用数据库编号唯一标识的分区中提供数据库中的缺省值。请参见“[在 UltraLite 中使用 GLOBAL AUTOINCREMENT](#)”一节第 119 页和“[UltraLite global\\_database\\_id 选项](#)”一节第 204 页。
  - **[ NOT ] NULL** 控制列是否可以包含 NULL。
  - **NEWID()** 生成一个唯一标识符值的函数。请参见“[NEWID 函数 \[Miscellaneous\]](#)”一节第 383 页。
  - **CURRENT TIMESTAMP** 组合 CURRENT DATE 和 CURRENT TIME，以构成包含年、月、日、小时、分钟、秒和秒的小数部分的 TIMESTAMP 值。秒的小数值存储到 3 个小数位。其准确性受系统时钟准确性的限制。请参见“[CURRENT TIMESTAMP 特殊值](#)”一节第 287 页。
  - **CURRENT DATE** 存储当前年月日。请参见“[CURRENT DATE 特殊值](#)”一节第 286 页。
  - **CURRENT TIME** 存储当前小时、分钟、秒和秒的小数值。请参见“[CURRENT TIME 特殊值](#)”一节第 286 页。
  - **constant-value** 列的数据类型的常量。该常量通常是一个数字或一个字符串。
  - **column-constraint 子句** 指定列约束以限制可出现在列中的值。列约束可以为下列项之一：
    - **PRIMARY KEY** 设置为 *column-constraint* 的一部分时，PRIMARY KEY 子句将此列设置为表的主键。主键用于唯一标识表中的每一行。缺省情况下，主键包括的列不能为 NULL。
    - **UNIQUE** 标识唯一标识表中各行的一列或多列。表中任何两行的值在所有指定的列中不能相同。表可以有多个唯一约束。不允许使用 NULL 值。

**table-constraint 子句** 指定表约束以限制此表中一个或多个列可包含的值。CONSTRAINT 子句用于为表约束指定标识符。表约束可以为主键约束、外键约束或唯一约束的形式，如下定义：

- **pkey-constraint 子句** 将指定列设置为表的主键。主键用于唯一标识表中的每一行。主键包括的列不能为 NULL。
- **fkey-constraint 子句** 指定外键约束以限制一个或多个列的值必须与另一个表的主键（或唯一约束）中的值匹配。
  - **NOT NULL 子句** 指定 NOT NULL 以在外键列中不允许使用 NULL。外键中的 NULL 表示主表中没有任何行与外表中的此行相对应。如果多列外键中至少有一个值为 NULL，则该键的其它列中可以保存的值没有限制。
  - **role-name 子句** 指定 *role-name* 命名外键。*role-name* 用于区分相同表中的外键。或者，可使用 CONSTRAINT *constraint-name* 命名外键。但是，请勿同时使用两种方法来命名外键。
  - **REFERENCES 子句** 指定 REFERENCES 子句以将主表中的一个或多个列定义为用作外键约束。在 REFERENCES 列约束中指定的所有 *column-name* 必须为主表中的列，必须受唯一约束或主键约束的制约。
  - **CHECK ON COMMIT** 强制执行外键约束之前，指定 CHECK ON COMMIT 以使数据库服务器等待 COMMIT。缺省情况下，在插入、更新或删除操作期间会立即强制执行外键约束。但是，设置了 CHECK ON COMMIT 时，即使数据库更改违反外键约束其也可按任意顺序进行，可以在出现下一个 COMMIT 之前更正不一致的数据。
- **unique-key-constraint 子句** 指定唯一约束来标识一个或多个唯一标识表中每行的列。表中任何两行的值在所有指定的列中不能相同。表可以有多个唯一约束。
- **WITH MAX HASH SIZE** 为此索引设置散列大小（按字节计）。该值替换对数据库有效的缺省 MaxHashSize 属性。要学习缺省大小，请参见“访问 UltraLite 数据库属性”一节第 199 页。

## 注释

除非约束引用表中的多列，否则通常会使用列约束。这些情况下，必须使用表约束。如果语句会导致违反约束，则该语句的执行不会完成。该语句在检测到错误前所做的任何更改都被撤消并报告错误。

表的每一行均有一个唯一主键值。

如果不指定角色名，则按如下方式分配角色名：

1. 如果没有与表名同名的角色名外键，则将表名指派为角色名。
2. 如果表名已被使用，则角色名为表名加上表的唯一三位零填充数字。

**模式更改** 如果模式更改同时开始，则语句将不会被释放。请参见“使用 DDL 语句更改模式”一节第 9 页。

当 CREATE TABLE 语句正在被处理时，UltraLite 不会处理引用该表的请求或查询。此外，当数据库包含活动查询或未提交的事务时，您也无法执行 CREATE TABLE 语句。

对于 UltraLite.NET 用户：除非也为所有数据对象（例如 ULDataReader）调用了 Dispose 方法，否则您将无法执行此语句。请参见“Dispose 方法”一节《UltraLite - .NET 编程》。

## 副作用

自动提交。

## 另请参见

- “UltraLite 中的表达式” 一节第 303 页
- “UltraLite DROP TABLE 语句” 一节第 448 页
- “CREATE TABLE 语句” 一节 《SQL Anywhere 服务器 - SQL 参考》
- “UltraLite 中的数据类型” 一节第 290 页
- “替换自动增量列的分区大小” 一节第 121 页

## 示例

下面的语句为图书馆数据库创建用于保存图书信息的表。

```
CREATE TABLE library_books (  
    isbn CHAR(20) PRIMARY KEY,  
    copyright_date DATE,  
    title CHAR(100),  
    author CHAR(50),  
    location CHAR(50),  
    FOREIGN KEY location REFERENCES room  
);
```

以下语句为图书馆数据库创建用于保存借出图书信息的表。date\_borrowed 的缺省值指示在创建条目的当天借出图书。date\_returned 列在归还图书前一直为 NULL。

```
CREATE TABLE borrowed_book (  
    loaner_name CHAR(100) PRIMARY KEY,  
    date_borrowed DATE NOT NULL DEFAULT CURRENT DATE,  
    date_returned DATE,  
    book CHAR(20)  
    FOREIGN KEY book REFERENCES library_books (isbn)  
);
```

下面的语句将为某销售数据库创建保存订单和订单项信息的表。

```
CREATE TABLE Orders (  
    order_num INTEGER NOT NULL PRIMARY KEY,  
    date_ordered DATE,  
    name CHAR(80)  
);  
CREATE TABLE Order_item (  
    order_num INTEGER NOT NULL,  
    item_num SMALLINT NOT NULL,  
    PRIMARY KEY (order_num, item_num),  
    FOREIGN KEY (order_num)  
    REFERENCES Orders (order_num)  
);
```

## UltraLite DELETE 语句

此语句用于从数据库中的表删除行。

### 语法

```
DELETE [ FROM ] table-name[[AS] correlation-name]  
[ WHERE search-condition ]
```

### 参数

**correlation-name** 在引用位于语句其它位置的表时使用的标识符。

**WHERE 子句** 如果指定了 WHERE 子句，则只删除满足 *search-condition* 的行。请参见“[UltraLite 中的搜索条件](#)”一节第 309 页。

WHERE 子句不支持非确定型函数（如 RAND）或变量。此子句也不限制列；当列在子查询中使用，可能需要引用另一个表。

### 注释

UltraLite 跟踪行状态的方式非常独特。请确定您理解删除和行状态的含义。请参见“[UltraLite 行状态](#)”一节第 12 页。

### 另请参见

- “[UltraLite START SYNCHRONIZATION DELETE 语句](#)”一节第 460 页
- “[UltraLite STOP SYNCHRONIZATION DELETE 语句](#)”一节第 461 页

### 示例

以下语句会从 Employees 表中删除雇员 105。

```
DELETE  
FROM Employees  
WHERE EmployeeID = 105;
```

以下语句会从 FinancialData 表中删除 2000 年之前的所有数据。

```
DELETE  
FROM FinancialData  
WHERE Year < 2000;
```

## UltraLite DROP INDEX 语句

此语句用于删除索引。

### 语法

```
DROP INDEX [ table-name. ] index-name
```

### 注释

不能删除表的主索引。

当 DROP INDEX 语句正在被处理时，UltraLite 不会处理引用该索引的请求或查询。此外，当数据库包含活动查询或未提交的事务时，您也无法执行 DROP INDEX 语句。

对于 UltraLite.NET 用户：除非也为所有数据对象（例如 ULDataReader）调用了 Dispose 方法，否则您将无法执行此语句。请参见“Dispose 方法”一节《UltraLite - .NET 编程》。

如果模式更改同时开始，则语句将不会被释放。请参见“使用 DDL 语句更改模式”一节第 9 页。

### 另请参见

- “UltraLite CREATE INDEX 语句”一节第 433 页
- “使用 UltraLite 索引”一节第 71 页

### 示例

以下语句在 FinancialData 表上删除虚构索引 fin\_codes\_idx:

```
DROP INDEX FinancialData.fin_codes_idx;
```

## UltraLite DROP PUBLICATION 语句

此语句用于删除发布。

### 语法

```
DROP PUBLICATION publication-name, ...
```

### 另请参见

- “设计 UltraLite 中的同步” 一节第 123 页
- “UltraLite ALTER PUBLICATION 语句” 一节第 425 页
- “UltraLite CREATE PUBLICATION 语句” 一节第 435 页

### 示例

以下语句删除 pub\_contact 发布。

```
DROP PUBLICATION pub_contact;
```



## UltraLite DROP SYNCHRONIZATION PROFILE 语句

此语句用于删除 UltraLite 同步配置文件。同步配置文件定义 UltraLite 数据库如何与 MobiLink 服务器同步。

### 语法

```
DROP SYNCHRONIZATION PROFILE sync-profile-name
```

### 参数

- **sync-profile-name** 同步配置文件的名称。

### 注释

无。

### 副作用

无。

### 另请参见

- [“UltraLite CREATE SYNCHRONIZATION PROFILE 语句”一节第 436 页](#)
- [“UltraLite ALTER SYNCHRONIZATION PROFILE 语句”一节第 426 页](#)
- [“UltraLite SYNCHRONIZE 语句”一节第 462 页](#)

### 示例

以下示例显示了用于删除名为 Test1 的同步配置文件的语法。

```
DROP SYNCHRONIZATION PROFILE Test1;
```

## UltraLite DROP TABLE 语句

此语句用于从数据库中删除表及其所有数据。

### 语法

**DROP TABLE** *table-name*

### 注释

DROP 语句从数据库中删除指定的表。表中的所有数据以及所有索引和键也均被删除。

当 DROP TABLE 语句正在被处理时，UltraLite 不会处理引用该表或其索引的请求或查询。此外，当有活动查询或未提交的事务时，您也无法执行 DROP TABLE 语句。

对于 UltraLite.NET，除非也为所有数据对象（例如 ULDataReader）调用了 Diapose 方法，否则您将无法执行此语句。请参见“[Dispose 方法](#)”一节《[UltraLite - .NET 编程](#)》。

如果模式更改同时开始，则语句将不会被释放。请参见“[使用 DDL 语句更改模式](#)”一节第 9 页。

### 另请参见

- [“UltraLite ALTER TABLE 语句”一节第 428 页](#)
- [“UltraLite CREATE TABLE 语句”一节第 440 页](#)

### 示例

以下语句从数据库中删除虚构表 EmployeeBenefits:

```
DROP TABLE EmployeeBenefits;
```

## UltraLite FROM 子句

此子句用于指定 SELECT 语句中涉及的表或视图。

### 语法

**FROM** *table-expression*, ...

*table-expression* :

```

 [ [ AS ] correlation-name ]
| ( select-list ) [ AS ] derived-table-name ( column-name, ... )
| ( table-expression )
| table-expression join-operator table-expression [ ON search-condition ] ...
```

*join-operator* :

```

,
| INNER JOIN
| CROSS JOIN
| LEFT OUTER JOIN
| JOIN
```

### 参数

**table-name** 基表或临时表。在 UltraLite 中，表不能由不同的用户拥有。如果用用户 ID 限定表，该 ID 将被忽略。

**correlation-name** 在引用位于语句其它位置的表时使用的标识符。例如，在下面语句中，a 被定义为 Contacts 表的相关名；b 为 Customers 表的相关名。

```

SELECT *
FROM Contacts a, Customers b
WHERE a.CustomerID=b.ID;
```

**derived-table-name** 派生表是在 FROM 子句中的嵌套 SELECT 语句。

派生表的选择列表中的项将由后面跟有句点 (.) 和列名 (可选) 的派生表名引用。如果列名非常明确，则可单独使用。

您不能在 SELECT 语句中引用派生表。请参见“表达式中的子查询”一节第 307 页。

**join-operator** 指定连接的类型。如果指定了逗号 (,) 或 CROSS JOIN，则不能指定 ON 子句。如果指定了 JOIN，则必须指定 ON 子句。对于 INNER JOIN 和 LEFT OUTER JOIN，ON 子句是可选的。

### 注释

当没有 FROM 子句时，SELECT 语句中的表达式必须是常量表达式。

#### 派生表

虽然此说明针对的是表，但它同样适用于派生表（除非另外说明）。

FROM 子句创建由所有指定表中的所有列组成的结果集。指定表中行的所有组合最初都在结果集中，但 JOIN 条件和/或 WHERE 条件通常会减少组合数。

如果未指定连接的类型，并且以逗号分隔列表的形式列出这些表，则缺省情况下会使用 CROSS JOIN。

对于 INNER 连接，使用 ON 子句或 WHERE 子句限制连接的结果会返回等效的结果。对于 OUTER 连接，使用 ON 或 WHERE 进行限制并不等效。

**注意**

UltraLite 不支持 KEY JOINS，也不支持 NATURAL 连接。

**另请参见**

- “连接：从多个表检索数据” 《SQL Anywhere 服务器 - SQL 的用法》
- “UltraLite DELETE 语句” 一节第 444 页
- “UltraLite SELECT 语句” 一节第 457 页
- “UltraLite UPDATE 语句” 一节第 467 页

**示例**

以下是有效的 FROM 子句：

```
...  
FROM Employees  
...  
  
...  
FROM Customers  
CROSS JOIN SalesOrders  
CROSS JOIN SalesOrderItems  
CROSS JOIN Products  
...
```

以下查询使用派生表在 Customers 表中返回在 SalesOrders 表中拥有超过三个订单的用户的姓名：

```
SELECT Surname, GivenName, number_of_orders  
FROM Customers JOIN  
    ( SELECT CustomerID, COUNT(*)  
      FROM SalesOrders  
      GROUP BY CustomerID )  
  AS sales_order_counts( CustomerID, number_of_orders )  
ON ( Customers.id = sales_order_counts.CustomerID )  
WHERE number_of_orders > 3;
```

## UltraLite INSERT 语句

此语句用于将行插入表中

### 语法

```
INSERT [ INTO ]
 [ ( column-name, ... ) ]
{ VALUES ( expression, ... ) | select-statement }
```

### 注释

INSERT 语句可用于插入单行或从查询结果集插入多行。

如果指定了列，则值将逐个插入到指定的列中。如果未指定列名的列表，则值将以它们出现在表中的顺序（与用 SELECT \* 检索的顺序相同）插入到表的列中。行以任意顺序插入到表中。

插入到表中的字符串始终以它们输入时采用的大小写进行保存，而不论数据库是否区分大小写。

### 另请参见

- [“UltraLite SELECT 语句”一节第 457 页](#)

### 示例

以下语句将 Eastern Sales 部门添加到数据库。

```
INSERT
INTO Departments ( DepartmentID, DepartmentName )
VALUES ( 230, 'Eastern Sales' );
```

## UltraLite LOAD TABLE 语句

此语句用于将批量数据从外部文件导入到数据库表中。此语句还提供对处理 SQL Anywhere 的 dbunload 实用程序 (reload.sql 文件) 输出的支持。

```
LOAD [ INTO ] TABLE [ owner.]table-name
(column-name, ...)
FROM stringfilename
[ load-option ... ]
```

*load-option* :

```
CHECK CONSTRAINTS { ON | OFF }
| COMPUTES { ON | OFF }
| DEFAULTS { ON | OFF }
| DELIMITED BY string
| ENCODING encoding
| ESCAPES { ON }
| FORMAT { ASCII | TEXT }
| ORDER { ON | OFF }
| QUOTES { ON | OFF }
| SKIP integer
| STRIP { ON | OFF | BOTH }
| WITH CHECKPOINT { ON | OFF }
```

*comment-prefix* : string

*encoding* : string

### 参数

- **column-name** 此子句用于指定要装载数据的一个或多个列。如果 DEFAULTS 选项为 OFF，则将列的列表中不存在的任何列都变为 NULL。如果 DEFAULTS 为 ON 并且列具有缺省值，则使用该值。如果 DEFAULTS 为 OFF 并且列列表中省略了不可为空的列，则数据库服务器会试图将空字符串转换为该列的类型。

如果指定了列列表，它会列出文件中应存在的列以及列的显示顺序。列名不能重复。

- **FROM string-filename** 此子句用于指定从中装载数据的文件。将 *string-filename* 以字符串的形式传递到数据库服务器。因此，该字符串遵循的数据库格式要求与其它 SQL 字符串一样。尤其是：

- 要指示目录路径，反斜线字符 (\) 必须用两个反斜线来表示。将数据从文件 *c:\temp\input.dat* 装载到 Employees 表的语句为：

```
LOAD TABLE Employees
FROM 'c:\\temp\\input.dat' ...
```

- 路径名相对于数据库服务器，而不是客户端应用程序。
- 可以使用 UNC 路径名从数据库服务器以外的计算机上的文件装载数据。

- **load-option 子句** 有几个装载选项可以指定，以控制数据是如何装载的。下面的列表给出了所支持的装载选项：

- **CHECK CONSTRAINTS 子句** 此子句控制装载期间是否检查约束。缺省情况下，CHECK CONSTRAINTS 为 ON，但是在 CHECK CONSTRAINTS 为 OFF 时，卸载实用程序 (ulunload) 可写出 LOAD TABLE 语句。将 CHECK CONSTRAINTS 设置为 OFF 禁用检查约束，这很有用，例如，在数据库重建过程。
- **COMPUTES 子句** 此选项正在被处理但是却被 UltraLite 忽略。
- **DEFAULTS 子句** 缺省情况下，DEFAULTS 被设置为 OFF。如果 DEFAULTS 为 OFF，则将任何未显示在列的列表中且可为空的列置为 NULL。如果将 DEFAULTS 设置为 OFF 并且列的列表中省略了不可为空的列，则数据库服务器会试图将空字符串转换为该列的类型。如果将 DEFAULTS 设置为 ON 并且列具有缺省值，则使用该值。
- **DELIMITED BY 子句** 此子句用于指定列分隔符字符串。缺省的列分隔符字符串是逗号；但是它可以是任何长度在 255 字节之内的字符串（如 ... DELIMITED BY '###' ...）。其它 SQL 字符串的格式化要求也适用。如果您要指定以制表符分隔的值，可使用制表符 (9) 的十六进制转义序列，... DELIMITED BY '\x09' ...。
- **ENCODING 子句** 此子句指定用于向数据库装载的数据的字符编码。
- **ESCAPES 子句** ESCAPES 始终为 ON，因此数据库服务器会识别反斜线字符后面的字符并将其解释为特殊字符。换行符可以作为组合 \n 包含在数据中，其它字符可以作为十六进制 ASCII 代码（如使用 \x09 代替制表符）包含在数据中。两个连续的反斜线字符 (\\) 被解释为单个反斜线。任何除 n、x、X 或 \ 以外的字符及前面的反斜线都被解释为两个单独的字符。例如，\q 插入反斜线和字母 q。
- **FORMAT 子句** 此子句指定正在从中装载数据的数据源的格式。使用 TEXT，假定输入行为字符（由 ENCODING 选项所定义），每个输入行占一行，值由列分隔符字符串分隔。还支持 ASCII。
- **QUOTES 子句** 此子句指定字符串是否用引号引起来。UltraLite 仅支持 ON，因此 LOAD TABLE 语句要求将字符串用引用符号引起来。此引号字符为撇号（单引号）。字符串中出现的第一个这样的字符将被视为该字符串的引号字符。字符串必须以匹配的引号结束。

列分隔符字符串可以包含在列值中。另外，假定引号字符不属于该值的一部分。因此，尽管地址中出现了逗号，以下行也会被视为两个值（而不是三个值）。此外，将地址引起来的引号不会插入到数据库中。

```
'123 High Street, Anytown', (715) 398-2354
```

要在值中包括引用符号，必须使用两个引号。以下这个行在第三列中包括一个单引号字符值：

```
'123 High Street, Anytown', '(715) 398-2354', ''
```

- **SKIP 子句** 此子句用于指定是否忽略文件开始位置的行。*integer* 参数指定要跳过的行数。例如，可以用此子句跳过包含列标题的行。
- **STRIP 子句** 此子句正在被处理但是却被忽略。此子句指定在插入无引号值之前，是否应去除这些值的前导或尾随空白。STRIP 选项接受以下选项：
  - **STRIP ON** 去除前导空白。
  - **STRIP OFF** 不去除前导空白或尾随空白。
  - **STRIP BOTH** 去除前导空白和尾随空白。

- **WITH CHECKPOINT 子句** 此子句用于指定是否执行检查点。缺省设置为 OFF。如果将此子句设置为 ON，则会在成功完成语句之后执行检查点操作。

## 注释

LOAD TABLE 允许执行从文件到数据库表的高效的大量插入。其主要提供对 SQL Anywhere dbunload 实用程序 (reload.sql 文件) 输出的支持。

仅有 Windows 和 Linux 支持 LOAD TABLE，而 Palm OS 或 Windows Mobile 则不支持。

使用 FORMAT TEXT 时，不指定任何值就表示 NULL。例如，如果需要三个值且文件包含 1,, 'Fred',, 则所插入的值为 1、NULL 和 Fred。如果文件包含 1,2,, 则会插入值 1、2 和 NULL。仅由空格组成的值也被视为 NULL 值。例如，如果文件包含 1, , 'Fred',, 则会插入值 1、NULL 和 Fred。所有其它值均被视为非 NULL。例如 " (单引号 单引号) 是一个空字符串。'NULL' 是一个包含四个字母的字符串。

如果由 LOAD TABLE 装载的列不允许使用 NULL 值，且文件值为 NULL，则会为数字列赋予值 0 (零)，为字符列赋予空字符串 (")。如果由 LOAD TABLE 装载的列允许使用 NULL 值，且文件值为 NULL，则列值将为 NULL (对于所有类型)。

如果表包含 a、b 和 c 三列，输入数据包含 a、b 和 c，而 LOAD 语句仅指定 a 和 b 列作为要将数据装入的列，则会向 c 列插入以下值：

- 如果将 DEFAULTS 指定为 ON 并且 c 列具有缺省值，则使用该缺省值。
- 如果 c 列没有缺省值，且列允许使用 NULL，则使用 NULL。
- 如果列 c 没有缺省值，且不允许使用 NULL，则使用零 (0) 或空字符串 (")，或者返回错误，这要根据列的数据类型而定。

## 副作用

自动提交。

## 另请参见

- [“UltraLite INSERT 语句”一节第 451 页](#)
- [“UltraLite 卸载旧数据库实用程序 \(ulunload\)”一节第 260 页](#)
- [“卸载实用程序 \(dbunload\)”一节《SQL Anywhere 服务器 - 数据库管理》](#)

## 标准和兼容性

- **SQL/2003** 服务商扩充。

## 示例

以下是 LOAD TABLE 的示例。首先，创建一个表，然后使用名为 *input.txt* 的文件将数据装载到表中。

```
CREATE TABLE t( a CHAR(100) primary key, let_me_default INT DEFAULT 1, c
CHAR(100) );
```

以下是名为 *input.txt* 的文件的内容：

```
'this_is_for_column_c', 'this_is_for_column_a', ignore_me
```



以下 LOAD 语句装载名为 *input.txt* 的文件:

```
LOAD TABLE T ( c, a ) FROM 'input.txt' FORMAT TEXT DEFAULTS ON;
```

命令 SELECT \* FROM t 生成结果集:

<b>a</b>	<b>let_me_default</b>	<b>c</b>
this_is_for_column_a	1	this_is_for_column_c

## UltraLite ROLLBACK 语句

此语句用于结束事务，并恢复自上次执行 COMMIT 或 ROLLBACK 语句以来对数据所做的任何更改。

### 语法

```
ROLLBACK [ WORK ]
```

### 注释

使用 UltraLite SQL 创建事务。事务包括自上次执行 ROLLBACK 或 COMMIT 以来所做的所有更改 (INSERT、UPDATE 和 DELETE)。ROLLBACK 语句可结束当前事务，并撤消自上一个 COMMIT 或 ROLLBACK 以来对该数据库所做的所有更改。

### 另请参见

- [“UltraLite COMMIT 语句”一节第 432 页](#)

### 示例

以下语句将数据库还原为上一次提交时其所处的状态：

```
ROLLBACK;
```

## UltraLite SELECT 语句

此语句用于从数据库中检索信息。

### 语法

```
SELECT [ DISTINCT ] [ row-limitation ]
select-list
[ FROM table-expression, ... ]
[ WHERE search-condition ]
[ GROUP BY group-by-expression, ... ]
[ ORDER BY order-by-expression, ... ]
[ FOR { UPDATE | READ ONLY } ]
[ OPTION ( FORCE ORDER ) ]
```

```
row-limitation :
FIRST
| TOP n [ START AT m ]
```

```
select-list :
expression [ [ AS ] alias-name ], ...
```

```
order-by-expression :
{ integer | expression } [ ASC | DESC ]
```

### 参数

**DISTINCT 子句** 指定 DISTINCT 以删除结果中重复的行。如果没有指定 DISTINCT，则返回满足 SELECT 语句子句的所有行（包括重复的行）。当指定 DISTINCT 时，很多语句的执行时间显著延长，因此只有在必要时才应使用 DISTINCT。

**row-limitation 子句** 使用行限制返回结果的子集。例如，指定 FIRST 来检索结果集的第一行。使用 TOP $n$  返回结果的第一  $n$  行。检索 TOP $n$  行时，指定 START AT $m$  来控制起始行的位置。为 SELECT 语句指定 ORDER BY 子句来对行进行排序，这样这些子句就可返回有意义的结果。

**select-list** 表达式列表用于指定从数据库中检索的内容。通常，选择列表中的表达式为列名。但是，它们可以是其它类型的表达式（如函数）。使用星号 (\*) 选择 FROM 子句中列出的所有表的所有列。或者，可以为每个在 *select-list* 中的表达式定义别名。使用别名可允许您从查询的其它位置（例如，从 WHERE 和 ORDER BY 子句中）引用 *select-list* 表达式。

**FROM 子句** 行是从 *table-expression* 指定的表和视图中检索的。请参见“UltraLite FROM 子句”一节第 449 页。

**WHERE 子句** 如果指定了 WHERE 子句，则只选择满足 *search-condition* 的行。请参见“UltraLite 中的搜索条件”一节第 309 页。

**GROUP BY 子句** 具有 GROUP BY 子句查询的结果对于 GROUP BY 表达式的每个不同的值集均包含一行。结果行通常称为组，因为对于表列表中的每组行，结果中均有一行。集合函数可应用于这些组中的行。如果 NULL 出现，则会被视为唯一值。

**ORDER BY 子句** 此子句根据子句中指定的表达式，对查询的结果进行排序。ORDER BY 子句中的每一个表达式均可以按升序排序 (ASC) 或降序排序 (DESC)（缺省值）存储。如果表达式是整数  $n$ ，则查询结果按选择列表中的第  $n$  个表达式排序。

确保以特定的顺序返回行的唯一方法是使用 ORDER BY。当缺少 ORDER BY 子句时，UltraLite 以最有效的顺序返回行。

**FOR 子句** 此子句具有两个控制查询行为的变量：

- **FOR READ ONLY** 此子句表示该查询并不用于更新。应尽量指定此子句，因为当已知查询不会用于更新时，UltraLite 有时能获得更好的性能。例如，当 UltraLite 懂得了只读权限是必需的，它就可以执行直接表扫描。FOR READ ONLY 是缺省行为。请参见“[使用直接页扫描](#)”一节第 108 页。
- **FOR UPDATE** 此子句允许将查询用于更新。必须显式指定该子句，否则不允许更新（FOR READ ONLY 是缺省行为）。

**OPTION (FORCE ORDER) 子句** 不建议将此子句用于通用用途。它会替换 UltraLite 的表的访问顺序的选择，并要求 UltraLite 以表在查询中的显示顺序来访问表。仅在确定查询顺序比 UltraLite 顺序更有效时，才会使用此子句。

UltraLite 也可使用执行计划来优化查询。请参见“[UltraLite 中的执行计划](#)”一节第 320 页。

## 注释

请始终记得关闭查询。否则，将无法释放内存，且保留的临时表的数量会发生不必要的增加。

## 另请参见

- [“UltraLite 性能与优化” 第 99 页](#)
- [“SELECT 语句” 一节 《SQL Anywhere 服务器 - SQL 参考》](#)
- [“查询数据” 《SQL Anywhere 服务器 - SQL 的用法》](#)

## 示例

以下语句从 Employees 表中选择雇员数。

```
SELECT COUNT(*)  
FROM Employees;
```

以下语句从 Employees 表中选择 10 行（从第 40 行开始到第 49 行结束）。

```
SELECT TOP 10 START AT 40 * FROM Employees;
```

## UltraLite SET OPTION 语句

此语句用于更改数据库选项的值。

### 语法

```
SET OPTION option-name=option-value
```

*option-name*: identifier

*option-value*: string, identifier, or number

### 注释

只能使用此语句设置数据库选项，并且创建数据库后不能修改属性。这些规则的例外是 **ml\_remote\_id**（请参见以下内容）。

您不能指定一个选项是否持久。选项在 UltraLite 中的实施方式决定其是持久选项还是临时选项。持久选项存储在 **sysuldata** 表中。仅在数据库停止运行之前才能使用临时选项。

唯一可以取消设置的数据库选项是 **ml\_remote\_id**。例如：

```
SET OPTION ml_remote_id=;
```

结果是 ID 被设置为 NULL。

### 另请参见

- “**sysuldata** 系统表” 一节第 276 页
- “UltraLite 数据库选项” 第 201 页
- “**DB\_PROPERTY** 函数 [System]” 一节第 355 页
- “UltraLite **ml\_remote\_id** 选项” 一节第 205 页

### 示例

以下语句将 **global\_database\_id** 选项设置为 100:

```
SET OPTION global_database_id=100;
```

## UltraLite START SYNCHRONIZATION DELETE 语句

此语句可用于为 MobiLink 同步重新启动有关已删除行的日志记录。

### 语法

```
START SYNCHRONIZATION DELETE
```

### 注释

UltraLite 数据库自动记录对需要同步的行所做的更改。UltraLite 在下次同步时将这些更改上传到统一数据库。此语句允许您重新启动之前由 STOP SYNCHRONIZATION DELETE 语句停止的有关已删除行的日志记录。

执行 STOP SYNCHRONIZATION DELETE 语句之后，不会同步在此连接上执行的任何删除操作。效果一直持续到执行 START SYNCHRONIZATION DELETE 语句为止。

如果您的应用程序未同步数据，则不要使用 START SYNCHRONIZATION DELETE。

UltraLite 跟踪行状态的方式非常独特。请确定您理解删除和行状态的含义。请参见“[UltraLite 行状态](#)”一节第 12 页。

### 另请参见

- “[UltraLite STOP SYNCHRONIZATION DELETE 语句](#)”一节第 461 页

### 示例

以下 SQL 语句序列阐释了如何使用 START SYNCHRONIZATION DELETE 和 STOP SYNCHRONIZATION DELETE。

```
STOP SYNCHRONIZATION DELETE;  
DELETE FROM PROPOSAL  
  WHERE last_modified < months( CURRENT_TIMESTAMP, -1 );  
START SYNCHRONIZATION DELETE;  
COMMIT;
```

## UltraLite STOP SYNCHRONIZATION DELETE 语句

此语句可用于为 MobiLink 同步停止有关已删除行的日志记录。

### 语法

**STOP SYNCHRONIZATION DELETE**

### 注释

UltraLite 数据库自动记录对需要同步的行所做的更改。UltraLite 在下次同步时将这些更改上载到统一数据库。此语句允许您停止之前使用 STOP SYNCHRONIZATION DELETE 语句启动的有关已删除行的日志记录。此命令对纠正远程数据库很有用，但要谨慎使用，因为它能有效地禁止 MobiLink 同步。您应该只是暂时停止有关删除的日志记录。

执行 STOP SYNCHRONIZATION DELETE 语句之后，不会同步在此连接上执行的任何删除操作。在执行 START SYNCHRONIZATION DELETE 语句之前始终有效。

如果您的应用程序不同步数据，则不要使用 STOP SYNCHRONIZATION DELETE。

UltraLite 跟踪行状态的方式非常独特。请确定您理解删除和行状态的含义。请参见“[UltraLite 行状态](#)”一节第 12 页。

### 另请参见

- [“UltraLite START SYNCHRONIZATION DELETE 语句”一节第 460 页](#)

### 示例

以下 SQL 语句序列阐释了如何使用 START SYNCHRONIZATION DELETE 和 STOP SYNCHRONIZATION DELETE。

```
STOP SYNCHRONIZATION DELETE;  
DELETE FROM PROPOSAL  
WHERE last modified < months( CURRENT TIMESTAMP, -1 );  
START SYNCHRONIZATION DELETE;  
COMMIT;
```

## UltraLite SYNCHRONIZE 语句

此语句用于通过 MobiLink 服务器同步 UltraLite 数据库。此同步根据同步配置文件中的参数或者可在语句本身中指定的参数进行配置。

### 语法

```
SYNCHRONIZE {  
  PROFILE sync-profile-name [ MERGE sync-option [ ;... ] ]  
  | USING sync-option [ ;... ]  
}
```

*sync-option* :  
*sync-option-name* = *sync-option-value*

*sync-option-name* : *string*

*sync-option-value* : *string*

### 参数

- **sync-profile-name** 同步配置文件的名称。
- **MERGE 子句** 当您想要添加或替换同步配置文件中所提供的选项时，使用此子句。
- **USING 子句** 当您想要指定同步选项而不引用同步配置文件时，使用此子句。
- **sync-option** 一个或多个同步选项值对的字符串用分号分隔。例如，'option1=value1;option2=value2'。
- **sync-option-name** 同步选项的名称。
- **sync-option-value** 同步选项的值。

### 注释

有关 UltraLite 所支持的同步配置文件选项的列表，请参见 [“UltraLite CREATE SYNCHRONIZATION PROFILE 语句”](#) 一节第 436 页。

要了解同步选项是如何在同步配置文件中与现有选项合并的，请参见 [“UltraLite ALTER SYNCHRONIZATION PROFILE 语句”](#) 一节第 426 页。

通过允许同步选项在其中合并，开发人员可以选择不在数据库中存储某些选项（例如，MobiLinkPwd）。

如果使用 UltraLite 定义和注册同步回调函数，则无论何时执行 SYNCHRONIZE 语句，都会将该同步的进度信息传递给回调函数。如果未注册任何回调，则将取消进度信息。

### 副作用

无。



## 另请参见

- “UltraLite ALTER SYNCHRONIZATION PROFILE 语句” 一节第 426 页
- “UltraLite DROP SYNCHRONIZATION PROFILE 语句” 一节第 447 页
- “ULRegisterSynchronizationCallback” 一节 《UltraLite - C 及 C++ 编程》

## 示例

以下示例显示了用于同步名为 Test1 的同步配置文件（其中，MobiLinkPwd 并未存储为配置文件的一部分）的语法：

```
SYNCHRONIZE PROFILE Test1 MERGE 'MobiLinkPwd=sql'
```

以下示例显示了用于向名为 Test1 的同步配置文件添加发布和 uploadonly 选项的语法。

```
SYNCHRONIZE PROFILE Test1  
MERGE 'publication=p2;uploadonly=on';
```

下面举例说明了如何使用 USING。

```
SYNCHRONIZE USING  
'MobiLinkUid=joe;MobiLinkPwd=sql;ScriptVersion=1;Stream=TCPIP{host=localhost  
}'
```

以下示例显示了用于同步发布和 uploadonly 选项的语法。

```
SYNCHRONIZE  
USING 'publication=p2;uploadonly=on';
```

## UltraLite TRUNCATE TABLE 语句

此语句用于删除表中的所有行，但不删除表。

### 语法

```
TRUNCATE TABLE table-name
```

### 注释

TRUNCATE TABLE 语句从表中删除所有行，并且在后续同步时，不会就删除这些行通知 MobiLink 服务器。它等效于执行以下语句：

```
STOP SYNCHRONIZATION DELETE;  
DELETE FROM TABLE;  
START SYNCHRONIZATION DELETE;
```

#### 注意

在同步或复制涉及的数据库上应谨慎使用该语句。因为没有通知 MobiLink 服务器，所以这种删除可导致出现不一致的情况，而这种不一致会引起同步或复制失败。

执行 TRUNCATE TABLE 语句后，表结构、所有索引以及约束和列定义仍然存在，仅删除了数据。

如果影响表的语句已经由其它请求或查询引用，则 TRUNCATE TABLE 语句将无法执行。同样，当表被变更时，UltraLite 不会处理引用表的请求。此外，当数据库包含活动查询或未提交的事务时，您也无法执行 TRUNCATE TABLE 语句。

对于 UltraLite.NET 用户：除非也为所有数据对象（例如 ULDataReader）调用了 Dispose 方法，否则您将无法执行此语句。请参见“[Dispose 方法](#)”一节《[UltraLite - .NET 编程](#)》。

**模式更改** 如果模式更改同时开始，则语句将不会被释放。请参见“[使用 DDL 语句更改模式](#)”一节第 9 页。

### 副作用

如果表包含的列被定义为 DEFAULT AUTOINCREMENT 或 DEFAULT GLOBAL AUTOINCREMENT，则 TRUNCATE TABLE 将重置该列的下一个可用值。

一旦行被标记为以 TRUNCATE TABLE 删除，除非用户发出 ROLLBACK 语句，否则执行此操作的用户将无法再访问它们。但是，它们仍然可以从其它连接进行访问。使用 COMMIT 进行永久删除，从而使这些数据不能从所有连接进行访问。

如果同步已截断的表，则所有应用于表的 INSERT 语句都将优先于 TRUNCATE TABLE 语句。

### 另请参见

- “[UltraLite DELETE 语句](#)”一节第 444 页
- “[UltraLite START SYNCHRONIZATION DELETE 语句](#)”一节第 460 页
- “[UltraLite STOP SYNCHRONIZATION DELETE 语句](#)”一节第 461 页

### 示例

以下语句从 Departments 表中删除所有行。

```
TRUNCATE TABLE Departments;
```

如果执行此示例，要确保执行 ROLLBACK 语句恢复您的更改。请参见“[UltraLite ROLLBACK 语句](#)”一节第 456 页。

## UltraLite UNION 语句

此语句用于合并两个或更多选择语句的结果。

### 语法

```
select-statement-without-ordering  
[ UNION [ ALL | DISTINCT ] select-statement-without-ordering ]...  
[ ORDER BY [ number [ ASC | DESC ], ... ]
```

### 注释

使用 UNION 可将多个 SELECT 语句的结果合并成一个较大的结果。每个 SELECT 语句在其各自的选择列表中必须有相同的表达式数，并且不能包含 ORDER BY 子句。

UNION ALL 的结果是合并联合的 SELECT 语句的结果。指定 UNION 或 UNION DISTINCT 以获取没有重复行的结果，但是，请注意，删除重复行增加了此语句的总执行时间。指定 UNION ALL 以允许重复行。

试图合并不同数据类型的相应表达式时，UltraLite 会尝试找到表示合并值的数据类型。如果这无法实现，则并集操作会失败并返回错误（例如，"无法将 'Surname' 转换为数字"）。

结果中显示的列名为第一个 SELECT 语句使用的列名（或别名）。

ORDER BY 子句使用整数建立顺序，整数指示对结果进行排序时所依据的一个（或多个）查询表达式。

### 另请参见

- [“UltraLite SELECT 语句”一节第 457 页](#)

### 示例

以下示例列出合并的 Employees 和 Customers 表中的所有不重复的姓。

```
SELECT Surname FROM Employees  
UNION  
SELECT Surname FROM Customers;
```

## UltraLite UPDATE 语句

此语句用于修改表中的行。

### 语法

```
UPDATE table-name[[AS] correlation-name]  
SET column-name = expression, ...  
[ WHERE search-condition ]
```

### 参数

**table-name** *table-name* 指定要更新的表的名称。仅允许指定一个表。

**correlation-name** 在引用位于语句其它位置的表时使用的标识符。

**SET 子句** 每个指定的列被设置为等号右侧表达式的值。对表达式没有任何限制。如果表达式为 *column-name*，则使用旧值。

只有在 SET 子句中指定的列才会更改其值。特别是，不能使用 UPDATE 将列的值设置为它的缺省值。

**WHERE 子句** 如果指定了 WHERE 子句，则只更新满足 *search-condition* 的行。请参见 [“UltraLite 中的搜索条件”](#) 一节第 309 页。

### 注释

UPDATE 语句修改表中的值。

插入到表中的字符串始终以它们输入时采用的大小写进行保存，而不论数据库是否区分大小写。

### 另请参见

- [“UltraLite INSERT 语句”](#) 一节第 451 页
- [“UltraLite DELETE 语句”](#) 一节第 444 页
- [“UltraLite 中的搜索条件”](#) 一节第 309 页

### 示例

以下语句将雇员 Philip Chin（雇员 129）从销售部转移到市场部（部门 400）。

```
UPDATE Employees  
SET DepartmentID = 400  
WHERE EmployeeID = 129;
```

使用 *correlation-name* 的一个实例。

```
UPDATE Employee E  
SET salary = salary * 1.05  
WHERE EXISTS( SELECT 1 FROM Sales S HAVING E.Sales > Avg( S.sales)  
GROUP BY S.dept_no )
```

---

---

# UltraLite 故障排除

## 目录

无法启动 UltraLite 引擎 .....	470
升级后无法连接数据库 .....	471
数据库损坏 .....	472
数据库大小不稳定 .....	473
将 ASCII 数据导入新数据库 .....	474
实用程序始终以前一版本运行 .....	475
结果集无法预测地更改 .....	476
UltraLite 引擎客户端错误，错误为 -764 .....	477

---

## 无法启动 UltraLite 引擎

### 症状

您已根据以下定义使用了 START 连接参数启动了 UltraLite 引擎；但客户端返回了 SQLE\_UNABLE\_TO\_CONNECT\_OR\_START。

```
START="\Program Files\uleng11.exe"
```

### 解释

引号的位置不正确。

### 建议

为使此参数工作，第一个引号后面必须跟随 \ 字符。例如，可以界定此路径中的空格，如下所示：

```
START=\ :Program Files\uleng11.exe"
```

或者

```
START="'\Program Files\uleng11.exe'"'
```



## 升级后无法连接数据库

### 症状

您已升级 UltraLite。您发现可以用管理工具创建空的 UltraLite 数据库。但是，当使用 Sybase Central 尝试连接此空数据库或其它任何 UltraLite 数据库时（包括 *CustDB.udb*），收到了错误消息。连接 SQL Anywhere 数据库时未发生意外。

### 解释

您没有关闭所有的 SQL Anywhere 应用程序和进程。所以，UltraLite 插件没有正确安装。

### 建议

删除 SQL Anywhere 然后重新安装。

1. 关闭 Sybase Central、Interactive SQL 和所有正在运行的数据库引擎。
2. 请运行以下命令：

```
dbisql -terminate
```

```
scjview -terminate
```

3. 打开 Windows [任务管理器]，然后结束所有的 *scjview.exe* 和 *dbisql.exe* 进程。
4. 重新安装最新版的 UltraLite。

### 另请参见

- “升级 UltraLite” 一节 《SQL Anywhere 11 - 更改和升级》

## 数据库损坏

### 症状

如果发生了以下情况您的数据库可能已经损坏：

- 发生了如下的错误：
  - SQLE\_DEVICE\_ERROR
  - SQLE\_DATABASE\_ERROR（也可以是其它问题的症状）
  - SQLE\_MEMORY\_ERROR（也可以是其它问题的症状）
- 崩溃或返回了无效的查询结果。

### 解释

有两种比较常见的数据库损坏原因：

- 如果设备使一些存在的问题存储在文件上，从而错误地更改了文件的内容，这是最常见的原因。这个问题通常会非常快地使数据库停止运行。
- 如果 UltraLite 代码中有错误因而没有正确维护索引，出现这种原因不太常见。因为对查询结果的更改难以察觉，所以很这些问题难发现会一直存在的时间更长。

### 建议

校验和用于检测脱机损坏情况，这有助于降低由于重要页面发生错误而损坏其它数据的可能性。如果校验和校验失败，则在数据库载入页面时，UltraLite 将立即停止数据库并报告致命错误。此错误无法更正。因而，您必须：

1. 将错误报告给 iAnywhere。如果您知道造成数据损坏发生事件的顺序，还知道这个错误是可重复的，这会很有用。
2. 如果您需要数据，请将数据库的内容卸载到文件。
3. 创建新的数据库。
4. 通过同步或通过装载已卸载的数据来重新填充数据。

### 另请参见

- [“UltraLite checksum\\_level 创建参数”一节第 168 页](#)
- [“ULSQLCode 枚举”一节《UltraLite - .NET 编程》](#)

## 数据库大小不稳定

### 症状

您的应用程序在多个客户端设备中收集了很多大二进制对象，并将此信息与统一数据库同步，然后从每个客户端删除同步的数据。然而，尽管数据已从数据库中删除，数据库的大小仍然很大。由于设备的资源有限应仔细管理文件的大小，这是个值得注意的问题。

### 解释

如果数据库中的数据增长那么数据库的大小只能增大。但是，数据增长结束后，数据库文件保持原来大小，不会自己减小。在文件内部保留可用空间。

### 建议

确保对未同步的表没有使用 `STOP SYNCHRONIZATION DELETE` 或 `TRUNCATE` 语句。而对未同步的表使用包含 `FROM table-name` 子句的 `DELETE` 语句。

同步后重新创建数据库：

1. 创建部署到各个设备的 UltraLite 数据库。
2. 创建定义客户端设备所需模式的 DDL 语句的 SQL 脚本。请参见“[部署 UltraLite 模式升级](#)”一节第 58 页。
3. 同步数据。
4. 删除数据库。
5. 创建新的空数据库，并通过使用 `ALTER DATABASE SCHEMA FROM FILE` 语句来使用标准的数据库模式。

### 另请参见

- “[UltraLite STOP SYNCHRONIZATION DELETE 语句](#)”一节第 461 页
- “[UltraLite TRUNCATE TABLE 语句](#)”一节第 464 页
- “[UltraLite DELETE 语句](#)”一节第 444 页
- “[UltraLite ALTER DATABASE SCHEMA FROM FILE 语句](#)”一节第 424 页

## 将 ASCII 数据导入新数据库

### 症状

您创建了一个新的 UltraLite 数据库，但有一个 csv ASCII 数据文件无法导入。

### 解释

任何 UltraLite 管理工具都不支持 csv 格式。

### 建议

可以尝试以下技术：

- 使用 Interactive SQL (dbisql) 导入数据。可以连接到 UltraLite 数据库，然后选择 [数据] » [Import Data]。或者，可以连接到 UltraLite 数据库，然后运行 INPUT 语句（此语句不能在 UltraLite PreparedStatement 对象中使用）。

#### 注意

UltraLite 需要主键。尽管 Interactive SQL 可以为您创建表，但它不能为表自动创建主键。出于此目始终连接您已创建的 UltraLite 数据库。

- 如果希望作为批处理过程的一部分合并此功能，您必须写出自己的代码。

### 另请参见

- “INPUT 语句 [Interactive SQL]” 一节 《SQL Anywhere 服务器 - SQL 参考》
- “UltraLite 的 Interactive SQL 实用程序 (dbisql)” 一节第 229 页

## 实用程序始终以前一版本运行

### 症状

您刚刚安装了 UltraLite 11。但是，当您尝试运行任何一个 UltraLite 实用程序时，始终是前一个版本启动。

### 解释

如果您的计算机上有多个版本的 UltraLite，在使用管理工具时必须注意系统路径。由于安装过程会将最近安装版本的可执行文件目录添加到系统路径的末尾，因此安装了该软件的新版本后，仍然有可能出现在无意中运行以前安装的版本的情况。

### 建议

对于这个问题有多种解决方案。请参见“使用实用程序”一节 [《SQL Anywhere 11 - 更改和升级》](#)。

## 结果集无法预测地更改

### 症状

您运行了一个查询，然而您预期的结果集在每次运行它时都更改。

### 解释

仔细查看您得到的结果集。结果集中结果确实不同吗？或者它们每次只是以最有效的顺序返回。所选的顺序每次执行查询时都更改，这取决于最近一次访问行的时间以及其它的因素。

### 建议

如果您的结果集必须以可预测的或一致的顺序返回，请确保 `SELECT` 语句中包含 `ORDER BY` 子句。如果结果集返回仍不正确，那么您的数据库可能已损坏。

### 另请参见

- [“UltraLite SELECT 语句”一节第 457 页](#)
- [“数据库损坏”一节第 472 页](#)

## UltraLite 引擎客户端错误，错误为 -764

### 适用于

Windows Mobile

### 症状

您正在 Windows Mobile 设备上运行 UltraLite 引擎，客户端返回了 -764 错误。

### 解释

-764 错误表示无法找到引擎，所以不能启动。

### 建议

请考虑以下的操作：

- 请考虑将引擎重新部署到推荐的部署位置，\Windows 目录。UltraLite 会在此位置自动查找引擎文件。
- 如果将引擎安装到其它任何位置，请确保连接代码使用了 START 连接参数。
- 如果您已使用了 START 连接参数，并确保引擎的路径正确，请确保对路径名中的特殊字符使用了正确的转义序列。

例如，在某些情况下，您可能需要更改此代码：

```
ULConnection conn = new ULConnection(@"dbf=\Program Files\HelloEngine
\HelloEngine.udb;
START=\Windows\uleng11.exe")
```

为类似的代码：

```
ULConnection conn = new ULConnection(@"dbf=\\\"Program Files \"\
\HelloEngine\HelloEngine.udb;
START=\\Windows\uleng11.exe");
```

### 另请参见

- “部署带 UltraLite 引擎的多个 UltraLite 应用程序” 一节第 49 页
- “UltraLite START 连接参数” 一节第 225 页

---



# 术语表

---

术语表 .....	481
-----------	-----



---

# 术语表

---

## Adaptive Server Anywhere (ASA)

SQL Anywhere Studio 的关系数据库服务器组件，专供在移动和嵌入式环境中使用，或作为中小型企业服务器使用。在版本 10.0.0 中，Adaptive Server Anywhere 更名为 SQL Anywhere 服务器，SQL Anywhere Studio 更名为 SQL Anywhere。

另请参见：[“SQL Anywhere”一节第 498 页](#)

## 包

Java 中相关类的集合。

## 被引用对象

一种对象（如表），该对象在另一个对象（如视图）的定义中被直接引用。

另请参见：[“主键”一节第 508 页](#)

## 编码

也称作字符编码，编码是一种方法，通过该方法可以将字符集中的每个字符映射到一个或多个字节的信息，这些信息通常以十六进制数字表示。编码的一个例子是 UTF-8。

另请参见：

- [“字符集”一节第 508 页](#)
- [“代码页”一节第 483 页](#)
- [“归类”一节第 487 页](#)

## 标识符

用于引用数据库对象（如表或列）的字符串。标识符可以包含 A 到 Z、a 到 z、0 到 9、下划线 (\_)、at 符号 (@)、数字符号 (#) 或美元符号 (\$) 中的任何字符。

## 并发

同时执行两个或更多个独立并且可能存在竞争关系的进程。SQL Anywhere 会自动使用锁定来隔离事务，并确保每个并发应用程序看到的数据集均一致。

另请参见：

- [“事务”一节第 495 页](#)
- [“隔离级别”一节第 486 页](#)

## 参考数据库

MobiLink 中一种用于 UltraLite 客户端开发的 SQL Anywhere 数据库。在开发过程中，可以将一个 SQL Anywhere 数据库同时作为参考数据库和统一数据库使用。通过其它产品建立的数据库无法用作参考数据库。

## 参照完整性

遵守数据一致性控制规则（具体而言，不同表中主键值与外键值之间的关系）。若要实现参照完整性，每个外键中的值必须与被引用表中行的主键值相符。

另请参见：

- “主键”一节第 508 页
- “外键”一节第 500 页

## 策略

QAnywhere 中指定应在何时进行消息传输的方式。

## 插件模块

Sybase Central 中一种用于访问和管理产品的方法。当您安装相应的产品时，插件通常会自动安装并注册 Sybase Central。通常，插件在 Sybase Central 主窗口中作为顶级容器出现，并且使用产品本身的名称，如 SQL Anywhere。

另请参见：“Sybase Central”一节第 499 页

## 查询

一条或一组 SQL 语句，用于访问和/或操作数据库中的数据。

另请参见：“SQL”一节第 498 页

## 冲突解决

在 MobiLink 中，冲突解决是指一种逻辑，它指定当两个用户修改不同远程数据库上同一行时的处理方法。

## 重定向器

一种 Web 服务器插件，用于为客户端与 MobiLink 服务器之间的请求和响应选择发送路径。此插件还实现了负荷平衡和故障转移机制。

## 抽取

SQL Remote 复制中从统一数据库卸载相应结构和数据的行为。此信息用于初始化远程数据库。

另请参见：“复制”一节第 485 页

---

## 触发器

一种特殊形式的存储过程，用户运行修改数据的查询时会自动执行该存储过程。

另请参见：

- [“行级触发器”一节第 487 页](#)
- [“语句级触发器”一节第 505 页](#)
- [“完整性”一节第 501 页](#)

## 传输规则

QAnywhere 中用于确定何时进行消息传输、传输哪些消息以及应在何时删除消息的逻辑。

## 窗口

作为分析功能执行对象的行组。一个窗口可以包含一行、多行或所有行的数据，这些数据已根据窗口定义中提供的分组规格进行了分区。窗口会进行移动，以包括为输入中的当前行执行计算所需的行数或行范围。窗口结构的主要优点是，不需要执行附加查询就可以有机会对结果进行分组和分析。

## 创建者 ID

UltraLite Palm OS 应用程序中一种在创建应用程序时指派的 ID。

## 存储过程

存储过程是数据库中存储的一组 SQL 指令，用于在数据库服务器上执行一组操作或查询。

## 代理表

一种本地表，它所包含的元数据可以像访问本地表一样访问远程数据库服务器上的表。

另请参见：[“元数据”一节第 506 页](#)

## 代理 ID

另请参见：[“客户端消息存储库 ID”一节第 491 页](#)

## 代码页

代码页是一种将字符集的字符映射到数字表示的编码，数字表示通常是 0 到 255 之间的一个整数。例如，Windows 代码页 1252 就是一个代码页。就本文档而言，代码页和编码这两个术语可以互换。

另请参见：

- [“字符集”一节第 508 页](#)
- [“编码”一节第 481 页](#)
- [“归类”一节第 487 页](#)

## DBA 权限

使用户能够在数据库中执行管理活动的权限级别。DBA 用户在缺省情况下具有 DBA 权限。

另请参见：[“数据库管理员 \(DBA\)” 一节第 497 页](#)

## dbspace

用于创建更多数据存储空间的附加数据库文件。一个数据库可以包含在最多 13 个独立的文件（一个初始文件和 12 个 dbspace）中。每个表及其索引必须包含在单个数据库文件中。SQL 命令 CREATE DBSPACE 可将新文件添加到数据库中。

另请参见：[“数据库文件” 一节第 498 页](#)

## 动态 SQL

执行前由程序以编程方式生成的 SQL。UltraLite 动态 SQL 是一种专用于小型设备的 SQL 变体。

## 对象树

Sybase Central 中数据库对象的层次。对象树的顶层显示您的 Sybase Central 版本所支持的全部产品。每种产品展开后会显示其自己的对象子树。

另请参见：[“Sybase Central” 一节第 499 页](#)

## EBF

快速错误修正软件。快速错误修正软件是含有一个或多个错误修正软件的软件子集。错误修正软件列在更新程序的发行说明中。错误修正软件更新可能只适用于具有相同版本号的已安装软件。已对该软件执行了一些测试，但该软件尚未进行完全测试。除非您自己已验证了软件的适用性，否则不要随应用程序分发这些文件。

## 发布

MobiLink 或 SQL Remote 中一种用于标识将要同步的数据的数据库对象。在 MobiLink 中，发布仅存在于客户端。一个发布包括多个项目。SQL Remote 用户可以通过预订发布来接收发布。MobiLink 用户可以通过创建发布的同步预订来同步发布。

另请参见：

- [“复制” 一节第 485 页](#)
- [“项目” 一节第 503 页](#)
- [“发布更新” 一节第 484 页](#)

## 发布更新

SQL Remote 复制中对一个数据库中的一个或多个发布所做更改的列表。发布更新将作为复制消息的一部分定期发送到远程数据库。

---

另请参见：

- “复制”一节第 485 页
- “发布”一节第 484 页

## 发布者

SQL Remote 复制中数据库内可以与其它复制数据库交换复制消息的单个用户。

另请参见：[“复制”一节第 485 页。](#)

## FILE

SQL Remote 复制中一种使用共享文件来交换复制消息的消息系统。它对测试以及在无显式消息传送系统的情况下进行的安装很有用。

另请参见[“复制”一节第 485 页。](#)

## 分析树

查询的代数表示。

## 服务

在 Windows 操作系统上，服务是在运行应用程序的用户 ID 未登录时的应用程序运行方式。

## 服务器管理请求

一种 QAnywhere 消息，其格式设置为 XML 并发送到 QAnywhere 系统队列，作为一种管理服务器消息存储库或监控 QAnywhere 应用程序的方法。

## 服务器启动的同步

一种从 MobiLink 服务器启动 MobiLink 同步的方式。

## 服务器消息存储库

QAnywhere 中在消息传输到客户端消息存储库或 JMS 系统之前服务器上用于临时存储消息的关系数据库。消息通过服务器消息存储库在各客户端之间进行交换。

## 复制

在物理上不相同的数据库之间共享数据。Sybase 有三种复制技术：MobiLink、SQL Remote 和复制服务器。

## 复制代理

请参见：[“LTM”一节第 492 页](#)

## 复制服务器

Sybase 的一种基于连接的复制技术，用于与 SQL Anywhere 和 Adaptive Server Enterprise 一起使用。它专用于在一些数据库之间进行接近实时的复制。

另请参见：[“LTM”一节第 492 页](#)

## 复制频率

SQL Remote 复制中一项针对每个远程用户的设置，它决定发布者消息代理向该远程用户发送复制消息的频率应为多少。

另请参见：[“复制”一节第 485 页](#)。

## 复制消息

SQL Remote 或复制服务器中一种在发布数据库与预订数据库之间发送的通信。消息包含复制系统所需的数据、直通语句及信息。

另请参见：

- [“复制”一节第 485 页](#)
- [“发布更新”一节第 484 页](#)

## 隔离级别

一个事务中的操作对其它并发事务中的操作的可见程度。隔离级别有四级，编号依次为 0 至 3。第 3 级提供最高级别的隔离。级别 0 为缺省设置。SQL Anywhere 还支持以下三个快照隔离级别：快照、语句快照和只读语句快照。

另请参见：[“快照隔离”一节第 491 页](#)

## 个人服务器

与客户端应用程序在同一台计算机上运行的数据库服务器。个人数据库服务器通常由单个用户在一台计算机上使用，但它可以支持来自该用户的几个并发连接。

## 工作表

一种内部存储区域，用于在查询优化过程中存储中间结果。

## 故障切换

在活动服务器、系统或网络出现故障或意外终止时切换到冗余或备用的服务器、系统或网络。故障转移会自动进行。

## 关系数据库管理系统 (RDBMS)

一种以相关表的形式存储数据的数据库管理系统。

另请参见：[“数据库管理系统 \(DBMS\)”一节第 497 页](#)



---

## 规范化

对数据库模式的改进，目的在于按照基于关系数据库理论的规则消除冗余并改善组织。

## 归类

定义数据库中文本属性的字符集与排序顺序的组合。对于 SQL Anywhere 数据库，缺省归类取决于运行服务器时所使用的操作系统和语言；例如，英语 Windows 系统上的缺省归类为 1252LATIN1。归类（也称作归类序列）用于对字符串进行比较和排序。

另请参见：

- “字符集”一节第 508 页
- “代码页”一节第 483 页
- “编码”一节第 481 页

## 行级触发器

每更改一行即执行一次的触发器。

另请参见：

- “触发器”一节第 483 页
- “语句级触发器”一节第 505 页

## 回退日志

对在每个未提交的事务执行过程中所做更改的记录。当收到 ROLLBACK 请求或者系统出现故障时，未提交的事务会从数据库中回退，将数据库返回其原先的状态。每个事务都有一个单独的回退日志，事务完成时日志会被删除。

另请参见：“事务”一节第 495 页

## iAnywhere JDBC 驱动程序

iAnywhere JDBC 驱动程序提供了一个 JDBC 驱动程序，与纯 Java jConnect JDBC 驱动程序相比，该驱动程序拥有一些性能优势和功能优点，但它不是纯 Java 解决方案。建议在大多数情况下使用 iAnywhere JDBC 驱动程序。

另请参见：

- “JDBC”一节第 488 页
- “jConnect”一节第 488 页

## InfoMaker

一种报告和数据维护工具，它用于创建复杂的表格、报告、图形、交叉表和表，并创建将这些报告用作构件块的应用程序。

## Interactive SQL

一种 SQL Anywhere 应用程序，用于查询和更改数据库中的数据以及修改数据库的结构。Interactive SQL 不但提供了一个用于输入 SQL 语句的窗格，还提供了一些用于返回有关查询处理过程的信息和结果集的窗格。

## JAR 文件

Java 档案文件。一种压缩的文件格式，由一个或多个用于 Java 应用程序的包的集合组成。它将安装和运行 Java 程序所需的全部资源都放在一个压缩文件中。

## Java 类

Java 中的主要代码结构单元。它是组合在一起的过程和变量的集合，将过程和变量组合在一起的原因是它们都与某个特定的可识别类别有关。

## jConnect

JavaSoft JDBC 标准的 Java 实现。它为 Java 开发人员提供多层和异类环境中的本地数据库访问。但在大多数情况下，iAnywhere JDBC 驱动程序是首选的 JDBC 驱动程序。

另请参见：

- [“JDBC”一节第 488 页](#)
- [“iAnywhere JDBC 驱动程序”一节第 487 页](#)

## JDBC

Java 数据库连接。一种 SQL 语言编程接口，它允许 Java 应用程序访问关系数据。首选的 JDBC 驱动程序是 iAnywhere JDBC 驱动程序。

另请参见：

- [“jConnect”一节第 488 页](#)
- [“iAnywhere JDBC 驱动程序”一节第 487 页](#)

## 基表

永久性的数据表。有时为区别于临时表和视图，会将这种表称作**基表**。

另请参见：

- [“临时表”一节第 491 页](#)
- [“视图”一节第 495 页](#)

## 基于会话的同步

一种同步类型，这种同步会使数据表示在统一数据库和远程数据库都一致。MobiLink 基于会话。

---

## 基于脚本的上载

MobiLink 中一种将上载过程自定义为使用日志文件的替代方法的方式。

## 基于 SQL 的同步

MobiLink 中一种使用 MobiLink 事件将表数据与支持 MobiLink 的统一数据库进行同步的方式。对于基于 SQL 的同步，可以直接使用 SQL，也可以使用面向 Java 和 .NET 平台的 MobiLink 服务器 API 返回 SQL。

## 基于文件的下载

在 MobiLink 中同步数据的一种方式，其中下载以文件的方式进行分发，从而支持脱机分发同步更改。

## 集成登录

一种登录功能，它允许将同一个用户 ID 和口令用于操作系统登录、网络登录和数据库连接。

## 监听器

一个程序 (dbsn)，用于 MobiLink 服务器启动的同步。监听器安装在远程设备上，它们被配置为在接收到来自通告程序的信息时启动针对设备的操作。

另请参见：[“服务器启动的同步”一节第 485 页](#)

## 检查点

将对数据库的所有更改都保存到数据库文件中的时间点。在其它时间，所提交的更改仅保存到事务日志中。

## 检查约束

对列或列集强制实施指定条件的一种限制。

另请参见：

- [“约束”一节第 507 页](#)
- [“外键约束”一节第 501 页](#)
- [“主键约束”一节第 508 页](#)
- [“唯一约束”一节第 502 页](#)

## 脚本

MobiLink 中为处理 MobiLink 事件而编写的代码。脚本通过编程方式控制数据交换，以满足业务需要。

另请参见：[“事件模型”一节第 495 页](#)

## 脚本版本

MobiLink 中为创建同步而一起应用的一组同步脚本。

## 校验

测试数据库、表或索引是否受到特定类型的文件损坏。

## 校验和

随数据库页本身一起记录的计算出的数据库页位数。校验和能够确保数据库页写入磁盘时位数相符，因此数据库管理系统可以通过它来验证数据库页的完整性。如果计数相符，即认为数据库页已成功写入。

## 镜像日志

另请参见：[“事务日志镜像”一节第 496 页](#)

## 角色

概念性数据库建模中从一个角度描述某种关系的动词或短语。您可以用两个角色来描述每种关系。例如，“包含”和“隶属于”便是角色。

## 角色名

外键的名称。由于它命名外表和主表之间的关系，因此称作角色名。缺省情况下，角色名就是表名，除非其它外键已经使用该名称（在这种情况下，缺省的角色名是表名后接一个三位的唯一数字）。也可以自己创建角色名。

另请参见：[“外键”一节第 500 页](#)

## 局部临时表

一种临时表，仅在复合语句执行期间或连接结束之前存在。当您只需要将数据集装载一次时，局部临时表非常有用。缺省情况下，行会在提交时被删除。

另请参见：

- [“临时表”一节第 491 页](#)
- [“全局临时表”一节第 494 页](#)

## 客户端/服务器

一种软件体系结构，在这种体系结构中，一个应用程序（客户端）从另一个应用程序（服务器）获取信息并向该应用程序发送信息。这两个应用程序常位于通过网络连接的不同计算机上。

## 客户端消息存储库

QAnywhere 中一种用于在远程设备上存储消息的 SQL Anywhere 数据库。

---

## 客户端消息存储库 ID

QAnywhere 中一种对客户端消息存储库进行唯一标识的 MobiLink 远程 ID。

## 快照隔离

一种为发出读请求的事务返回数据的已提交版本的隔离级别。SQL Anywhere 提供了以下三种快照隔离级别：快照、语句快照和只读语句快照。使用快照隔离时，读操作不会阻塞写操作。

另请参见：[“隔离级别”一节第 486 页](#)

## 连接

关系系统中的一种基本操作，它通过比较指定列中的值将两个或更多个表中的行链接在一起。

## 连接 ID

用于标识客户端应用程序与数据库之间给定连接的唯一编号。可以使用以下 SQL 语句来确定当前连接 ID：

```
SELECT CONNECTION_PROPERTY( 'Number' );
```

## 连接类型

SQL Anywhere 提供了四种类型的连接：交叉连接、键连接、自然连接和使用 ON 子句的连接。

另请参见：[“连接”一节第 491 页](#)

## 连接配置

连接到数据库所需的一组参数，如用户名、口令和服务器名称，它们在存储后即可方便地使用。

## 连接启动的同步

一种 MobiLink 服务器启动的同步，在这种同步下，连接发生变化时会启动同步。

另请参见：[“服务器启动的同步”一节第 485 页](#)

## 连接条件

一种影响连接结果的限制。您可以通过紧跟在连接语句的后面插入 ON 子句或 WHERE 子句来指定连接条件。对于自然连接和关键连接，SQL Anywhere 会生成连接条件。

另请参见：

- [“连接”一节第 491 页](#)
- [“生成的连接条件”一节第 496 页](#)

## 临时表

为临时存储数据而创建的表。有两种类型：全局临时表和局部临时表。

另请参见：

- [“局部临时表”一节第 490 页](#)
- [“全局临时表”一节第 494 页](#)

## LTM

日志传送管理器（Log Transfer Manager，简称 LTM）也称作复制代理。LTM 是一个与 Replication Server 一起使用的程序，它读取数据库事务日志并将提交的更改发送到 Sybase 复制服务器。

请参见：[“复制服务器”一节第 486 页](#)

## 轮询

在 MobiLink 服务器启动的同步中，轻量级轮询器（例如 MobiLink 监听器）从通告程序请求推式通知的方式。

另请参见：[“服务器启动的同步”一节第 485 页](#)

## 逻辑索引

指向物理索引的引用（指针）。磁盘上不存储逻辑索引的索引结构。

## 命令文件

包含 SQL 语句的文本文件。命令文件可以手工建立，也可以通过数据库实用程序自动建立。例如，dbunload 实用程序会创建一个命令文件，其中包含重新创建给定数据库所需的 SQL 语句。

## MobiLink

一种基于会话的同步技术，其设计用途是将 UltraLite 和 SQL Anywhere 远程数据库与统一数据库同步。

另请参见：

- [“统一数据库”一节第 499 页](#)
- [“同步”一节第 499 页](#)
- [“UltraLite”一节第 500 页](#)

## MobiLink 服务器

运行 MobiLink 同步的计算机程序，即 mlsrv11。

## MobiLink 监控器

一种用于监控 MobiLink 同步的图形化工具。

---

## MobiLink 客户端

有两种 MobiLink 客户端。对于 SQL Anywhere 远程数据库，MobiLink 客户端是 dbmlsync 命令行实用程序。对于 UltraLite 远程数据库，MobiLink 客户端内置于 UltraLite 运行时库中。

## MobiLink 系统表

MobiLink 同步所需的系统表。它们由 MobiLink 安装程序脚本安装到 MobiLink 统一数据库中。

## MobiLink 用户

MobiLink 用户用于与 MobiLink 服务器进行连接。在远程数据库上创建 MobiLink 用户，然后在统一数据库中注册该用户。MobiLink 用户名完全独立于数据库用户名。

## 模式

数据库的结构，其中包括表、列和索引以及它们之间的关系。

## 内连接

一种连接，在这种连接中，仅当两个表都满足连接条件时才会出现在结果集中。内连接是缺省设置。

另请参见：

- [“连接”一节第 491 页](#)
- [“外连接”一节第 501 页](#)

## ODBC

开放式数据库连接。一种用于与数据库管理系统连接的标准 Windows 接口。ODBC 是 SQL Anywhere 所支持的几种接口之一。

## ODBC 管理器

一种随 Windows 操作系统提供的 Microsoft 程序，用于设置 ODBC 数据源。

## ODBC 数据源

用户要通过 ODBC 访问的数据的规范以及获取该数据时所需的信息。

## PDB

Palm 数据库文件。

## PowerDesigner

一种数据库建模应用程序。PowerDesigner 为设计数据库或数据仓库提供了结构化的方法。SQL Anywhere 包括 PowerDesigner 的 Physical Data Model 组件。

## PowerJ

一种 Sybase 产品，用于开发 Java 应用程序。

## QAnywhere

应用程序到应用程序的消息传递（包括移动设备到移动设备和移动设备与企业之间的消息传递），它使在移动或无线设备上运行的自定义程序能够与处在中央位置的服务器应用程序进行通信。

## QAnywhere 代理

QAnywhere 中一种运行在客户端设备上的进程，用于监控客户端消息存储库和确定应在何时传输消息。

## 嵌入式 SQL

一种 C 语言程序编程接口。SQL Anywhere 嵌入式 SQL 是 ANSI 和 IBM 标准的实现。

## 轻量级轮询器

在 MobiLink 服务器启动的同步中，轮询来自 MobiLink 服务器的推式通知的设备应用程序。

另请参见：[“服务器启动的同步”一节第 485 页](#)

## 全局临时表

一种临时表，在被显式地删除之前，其数据定义对所有用户都可见。全局临时表允许用户各自打开一个表的相同实例。缺省情况下，行在提交时被删除，并且始终是在连接结束时被删除。

另请参见：

- [“临时表”一节第 491 页](#)
- [“局部临时表”一节第 490 页](#)

## 日志文件

SQL Anywhere 所维护的事务日志。该日志文件用于确保在出现系统或介质故障时可以恢复数据库、提高数据库性能以及使用 SQL Remote 实现数据复制。

另请参见：

- [“事务日志”一节第 495 页](#)
- [“事务日志镜像”一节第 496 页](#)
- [“完全备份”一节第 501 页](#)

## 散列

散列是一种将索引条目转化为键的索引优化。索引散列旨在通过将足够的行实际数据与其行 ID 包括在一起，以避免进行先查找行、后装载行然后再将行解出才能得出索引值的高开销操作。



---

## 上载

同步过程的一个阶段，在此阶段数据从远程数据库传送到统一数据库。

## 设备跟踪

在 MobiLink 服务器启动的同步中，允许使用标识设备的 MobiLink 用户名来对消息进行寻址的功能。

另请参见：[“服务器启动的同步”一节第 485 页](#)

## 实例化视图

实例化视图是指已计算并已存储在磁盘上的视图。实例化视图同时具有视图的特征（使用查询说明进行定义）和表的特征（可以对其执行大多数表操作）。

另请参见：

- [“基表”一节第 488 页](#)
- [“视图”一节第 495 页](#)

## 世代号

MobiLink 中的一种机制，用于强制远程数据库先上载数据，然后再应用任何其它下载文件。

另请参见：[“基于文件的下载”一节第 489 页](#)

## 事件模型

MobiLink 中组成同步的事件（如 `begin_synchronization` 和 `download_cursor`）序列。如果为事件创建了脚本，则会调用事件。

## 视图

一种作为对象存储在数据库中的 `SELECT` 语句。它使用户能够看到一个或多个表中的行子集或列子集。每当用户使用特定表或表组合的视图时，都将利用存储在这些表中的信息重新计算视图。视图对确保安全以及定制数据库信息的外观来使数据访问简单明了有帮助。

## 事务

组成一个逻辑工作单元的 `SQL` 语句序列。事务要么全部得到处理，要么根本不做处理。`SQL Anywhere` 支持事务处理，并内置了锁定功能，使并发事务能够访问数据库而又不损坏数据。事务要么以 `COMMIT` 语句结束，该语句使对数据的更改成为永久性更改；要么以 `ROLLBACK` 语句结束，该语句撤消在事务执行过程中所做的全部更改。

## 事务日志

一种按进行更改的顺序存储对数据库所做全部更改的文件。它会提高性能并支持在数据库文件损坏时恢复数据。

## 事务日志镜像

同时维护的事务日志文件的完全相同副本（可选）。每当数据库更改写入事务日志文件时，也会同时写入事务日志镜像文件。

镜像文件应与事务日志保留在不同的设备上，这样在任意设备出现故障时，日志的其它副本会确保数据可以安全地恢复。

另请参见：[“事务日志”一节第 495 页](#)

## 事务完整性

MobiLink 中对整个同步系统事务的有保证维护。要么同步整个事务，要么不对事务的任何部分进行同步。

## 生成的连接条件

一种自动生成的对连接结果的限制。有两种类型：关键和自然。指定 KEY JOIN 或指定关键字 JOIN 但不使用关键字 CROSS、NATURAL 或 ON 时，会生成关键连接。对于关键连接，所生成的连接条件取决于表之间的外键关系。指定 NATURAL JOIN 时会生成自然连接；所生成的连接条件基于两个表中的公用列名。

另请参见：

- [“连接”一节第 491 页](#)
- [“连接条件”一节第 491 页](#)

## 受保护的功能

数据库服务器启动时由 -sf 选项指定的功能，该数据库服务器上运行的任何数据库都无法使用该功能。

## 授权选项

一种权限级别，它允许用户向其他用户授予权限。

## 数据操作语言 (DML)

用于操作数据库中数据的 SQL 语句子集。DML 语句可以检索、插入、更新和删除数据库中的数据。

## 数据定义语言 (DDL)

用于定义数据库中数据结构的 SQL 语句子集。DDL 语句可以创建、修改和删除数据库对象（如表和用户）。

## 数据类型

数据的格式，如 CHAR 或 NUMERIC。在 ANSI SQL 标准中，数据类型也可以包括对大小、字符集和归类的限制。

---

另请参见：[“域”一节第 505 页](#)

## 数据立方体

一种多维结果集，每一维都以不同的方式对相同的结果进行分组和排序。数据立方体提供了有关数据的综合性信息，如果不使用数据立方体，要获得同样的信息就必须进行自连接查询和相关子查询。数据立方体是 OLAP 功能的一部分。

## 数据库

通过主键和外键关联的表的集合。表包含数据库中的信息。表和键一起定义数据库的结构。数据库管理系统会访问此信息。

另请参见：

- [“外键”一节第 500 页](#)
- [“主键”一节第 508 页](#)
- [“数据库管理系统 \(DBMS\)”一节第 497 页](#)
- [“关系数据库管理系统 \(RDBMS\)”一节第 486 页](#)

## 数据库对象

包含或接收信息的数据库组件。表、索引、视图、过程和触发器便是数据库对象。

## 数据库服务器

对所有针对数据库信息的访问进行管理的计算机程序。SQL Anywhere 提供了两种类型的服务器：网络服务器和个人服务器。

## 数据库管理系统 (DBMS)

用于创建和使用数据库的程序的集合。

另请参见：[“关系数据库管理系统 \(RDBMS\)”一节第 486 页](#)

## 数据库管理员 (DBA)

具有维护数据库所需权限的用户。DBA 通常负责对数据库模式的所有更改以及管理用户和组。数据库管理员角色自动内置于数据库中，其用户 ID 为 DBA，口令是 sql。

## 数据库连接

客户端应用程序与数据库之间的通信渠道。必须具有有效的用户 ID 和口令才能建立连接。为用户 ID 授予的特权决定了在连接过程中可以执行的操作。

## 数据库名称

服务器装载数据库时为数据库指定的名称。缺省数据库名是初始数据库文件的文件名（不含扩展名）。

另请参见：[“数据库文件”一节第 498 页](#)

## 数据库所有者 (dbo)

一种特殊的用户，他拥有不归 SYS 所有的系统对象。

另请参见：

- “数据库管理员 (DBA)” 一节第 497 页
- “SYS” 一节第 499 页

## 数据库文件

数据库保存在一个或多个数据库文件中。其中一个为初始文件，后面的文件称作 `dbspace`。每个表（包括其索引）都必须包含在单个数据库文件中。

另请参见：“`dbspace`” 一节第 484 页

## 死锁

一组事务会进入的一种特殊状态，在该状态下这些事务都不能继续执行。

## SQL

用于与关系数据库进行通信的语言。ANSI 定义了 SQL 的标准，其最新标准是 SQL-2003。SQL 的非官方全称是结构化查询语言。

## SQL Anywhere

SQL Anywhere 的关系数据库服务器组件，专供在移动和嵌入式环境中使用，或作为中小型企业的服务器使用。SQL Anywhere 也是包含 SQL Anywhere RDBMS、UltraLite RDBMS、MobiLink 同步软件和其它组件的软件包的名称。

## SQL Remote

一种基于消息的数据复制技术，用于在统一数据库与远程数据库之间进行双向复制。统一数据库和远程数据库必须是 SQL Anywhere。

## SQL 语句

包含用于将指令传递给 DBMS 的 SQL 关键字的字符串。

另请参见：

- “模式” 一节第 493 页
- “SQL” 一节第 498 页
- “数据库管理系统 (DBMS)” 一节第 497 页

## 锁定

一种在同时执行多个事务的过程中保护数据完整性的并发控制机制。SQL Anywhere 会自动应用锁以防止两个连接同时更改同一数据，并防止其它连接读取正接受更改的数据。

您可以通过设置隔离级别来控制锁定。

---

另请参见：

- [“隔离级别”一节第 486 页](#)
- [“并发”一节第 481 页](#)
- [“完整性”一节第 501 页](#)

## 索引

一组已排序的、与基表中的一个或多个列关联的键和指针。在表中一个或多个列上设置索引可以提高性能。

## Sybase Central

一种数据库管理工具，通过图形用户界面提供 SQL Anywhere 数据库设置、属性和实用程序。Sybase Central 也可用于管理其它 Sybase 产品，其中包括 MobiLink。

## SYS

一种拥有大多数系统对象的特殊用户。无法以 SYS 身份登录。

## 统一数据库

在分布式数据库环境中，是指用于存储数据主副本的数据库。出现冲突或差异时，将把统一数据库视为具有数据的主副本。

另请参见：

- [“同步”一节第 499 页](#)
- [“复制”一节第 485 页](#)

## 通信流

MobiLink 中 MobiLink 客户端与 MobiLink 服务器之间进行通信时所使用的网络协议。

## 通告程序

一种由 MobiLink 服务器启动的同步使用的程序。通告程序集成在 MobiLink 服务器中。它们会检查统一数据库是否有推式请求，并发送推式通知。

另请参见：

- [“服务器启动的同步”一节第 485 页](#)
- [“监听器”一节第 489 页](#)

## 同步

利用 MobiLink 技术在数据库之间复制数据的过程。

在 SQL Remote 中，同步专指以初始数据集初始化远程数据库的过程。

另请参见：

- [“MobiLink”一节第 492 页](#)
- [“SQL Remote”一节第 498 页](#)

### 推式请求

在 MobiLink 服务器启动的同步中，通告程序通过检查它来确定推式通知是否需要发送到设备的结果集中的一行值。

另请参见：[“服务器启动的同步”一节第 485 页](#)

### 推式通知

QAnywhere 中一种从服务器传送到 QAnywhere 客户端的特殊消息，用于提示客户端启动消息传输。在 MobiLink 服务器启动的同步中，从通告程序传送到包含推式请求数据和内部信息的设备的特殊消息。

另请参见：

- [“QAnywhere”一节第 494 页](#)
- [“服务器启动的同步”一节第 485 页](#)

### UltraLite

一种针对小型设备、移动设备和嵌入式设备进行了优化的数据库。所面向的平台包括手机、传呼机和个人记事本。

### UltraLite 运行时

一种过程中关系数据库管理系统，其中包括一个内置 MobiLink 同步客户端。每个 UltraLite 编程接口使用的库以及 UltraLite 引擎中都包括 UltraLite 运行时。

### 外表

包含外键的表。

另请参见：[“外键”一节第 500 页](#)

### 外部登录

与远程服务器通信时使用的替代登录名和口令。缺省情况下，SQL Anywhere 每次代表其客户端连接到远程服务器时都会使用这些客户端的名称和口令。但是，您可以通过创建外部登录来替换这一缺省设置。外部登录是指与远程服务器通信时使用的替代登录名和口令。

### 外键

一个表中复制另一个表中主键值的一个或多个列。外键建立表间的关系。

---

另请参见：

- [“主键”一节第 508 页](#)
- [“外表”一节第 500 页](#)

## 外键约束

对单个列或一组列的限制，指定表中的数据与某个其它表中数据的关系。对列集施加外键约束可使这些列成为外键。

另请参见：

- [“约束”一节第 507 页](#)
- [“检查约束”一节第 489 页](#)
- [“主键约束”一节第 508 页](#)
- [“唯一约束”一节第 502 页](#)

## 外连接

一种保留表中所有行的连接。SQL Anywhere 支持左、右和完全外连接。左外连接保留表中位于连接运算符左侧的行，当右表中的行不满足连接条件时，它将返回空值。完全外连接保留两个表中的所有行。

另请参见：

- [“连接”一节第 491 页](#)
- [“内连接”一节第 493 页](#)

## 完全备份

对整个数据库和事务日志（可选）的备份。完全备份包含数据库中的所有信息，因此可以在系统或介质出现故障时提供保护。

另请参见：[“增量备份”一节第 507 页](#)

## 完整性

遵守完整性规则的情况，完整性规则确保数据正确并准确，而且数据库的关系结构保持不变。

另请参见：[“参照完整性”一节第 482 页](#)

## 网关

一种 MobiLink 对象，存储在 MobiLink 系统表或通告程序属性文件中，包含有关如何发送用于服务器启动同步的消息的信息。

另请参见：[“服务器启动的同步”一节第 485 页](#)

## 网络服务器

从共享公共网络的计算机接受连接的数据库服务器。

另请参见：[“个人服务器”一节第 486 页](#)

## 网络协议

通信类型，如 TCP/IP 或 HTTP。

## 维护版本

维护版本是一套完整的软件，它升级已安装的具有相同主版本号的较早版本的软件（版本号格式是 *major.minor.patch.build*）。升级程序的发行说明中列出了错误修正软件和其它更改。

## 唯一约束

对某个列或一组列的限制，它要求所有非空值都各不相同。一个表可以有多个唯一约束。

另请参见：

- [“外键约束”一节第 501 页](#)
- [“主键约束”一节第 508 页](#)
- [“约束”一节第 507 页](#)

## 谓语句

一种条件表达式，可以选择性地将其与逻辑运算符 AND 和 OR 组合在一起，以组成 WHERE 或 HAVING 子句中的条件集。在 SQL 中，求值结果为 UNKNOWN 的谓语句将解释为 FALSE。

## 位数组

位数组是一种用于有效率地存储位序列的数组数据结构。位数组与字符串类似，不同的是其各个部分由 0（零）和 1（一）而不是字符组成。位数组通常用于保存一串布尔值。

## Windows

Microsoft Windows 操作系统系列，如 Windows Vista、Windows XP 和 Windows 200x。

## Windows CE

请参见 [“Windows Mobile”一节第 502 页](#)。

## Windows Mobile

Microsoft 为移动设备制造的操作系统系列。

## 文件定义数据库

MobiLink 中一种用于创建下载文件的 SQL Anywhere 数据库。

另请参见：[“基于文件的下载”一节第 489 页](#)



---

## 物理索引

索引存储在磁盘上的实际索引结构。

## 系统表

一种表，由 SYS 或 dbo 拥有，用于保存元数据。系统表也称作数据字典表，由数据库服务器创建并维护。

## 系统对象

由 SYS 或 dbo 拥有的数据库对象。

## 系统视图

存在于每一个数据库中的一种视图，它以易于理解的格式表示系统表中包含的信息。

## 下载

同步过程的一个阶段，在此阶段数据从统一数据库传送到远程数据库。

## 相关名

查询的 FROM 子句中使用的表或视图的名称—要么是表或视图的原始名称，要么是在 FROM 子句中定义的替代名称。

## 项目

在 MobiLink 或 SQL Remote 中，项目是表示整个表或表中行和列子集的数据库对象。项目在发布中组合在一起。

另请参见：

- [“复制”一节第 485 页](#)
- [“发布”一节第 484 页](#)

## 消息存储库

QAnywhere 中客户端和服务器设备上存储消息的数据库。

另请参见：

- [“客户端消息存储库”一节第 490 页](#)
- [“服务器消息存储库”一节第 485 页](#)

## 消息类型

SQL Remote 复制中指定远程用户与统一数据库发布者通信方式的数据库对象。一个统一数据库可能定义了几种消息类型，这样一来，不同的远程用户就可以使用不同的消息系统与统一数据库进行通信。

另请参见：

- [“复制”一节第 485 页](#)
- [“统一数据库”一节第 499 页](#)

### 消息日志

可存储来自数据库服务器或 MobiLink 服务器等应用程序的消息的日志。此类信息还可以出现在消息窗口中或记录到文件中。消息日志包括信息性消息、错误、警告以及来自 MESSAGE 语句的消息。

### 消息系统

SQL Remote 复制中用于在统一数据库与远程数据库之间交换消息的协议。SQL Anywhere 包括对以下消息系统的支持：FILE、FTP 和 SMTP。

另请参见：

- [“复制”一节第 485 页](#)
- [“FILE”一节第 485 页](#)

### 卸载

卸载数据库时会将数据库的结构和/或数据导出到文本文件（如果是结构，则导出到 SQL 命令文件中；如果是数据，则导出到 ASCII 逗号分隔文件中）。使用卸载实用程序来卸载数据库。

此外，您也可以使用 UNLOAD 语句卸载数据的选定部分。

### 性能统计

反映数据库系统性能的值。例如，CURRREAD 统计表示数据库服务器已发出但尚未完成的文件读取次数。

### 业务规则

基于实际要求的准则。通常，业务规则通过检查约束、用户定义数据类型以及事务的正确使用来实现。

另请参见：

- [“约束”一节第 507 页](#)
- [“用户定义数据类型”一节第 505 页](#)

### 引用对象

一种对象（如视图），其定义直接引用数据库中的另一个对象（如表）。

另请参见：[“外键”一节第 500 页](#)

---

## 用户定义数据类型

请参见“域”一节第 505 页。

## 游标

指向结果集的已命名链接，用于通过编程接口访问和更新行。在 SQL Anywhere 中，游标支持在查询结果中进行向前和向后移动。游标由两部分组成：游标结果集（通常由 SELECT 语句定义）和游标位置。

另请参见：

- “游标结果集”一节第 505 页
- “游标位置”一节第 505 页

## 游标结果集

与游标关联的查询所得到的行集。

另请参见：

- “游标”一节第 505 页
- “游标位置”一节第 505 页

## 游标位置

指向游标结果集中一个行的指针。

另请参见：

- “游标”一节第 505 页
- “游标结果集”一节第 505 页

## 语句级触发器

在整个触发语句完成后执行的触发器。

另请参见：

- “触发器”一节第 483 页
- “行级触发器”一节第 487 页

## 域

内置数据类型的别名，其中包括适用的精度值和小数位值，还可以选择是否包括 DEFAULT 值和 CHECK 条件。SQL Anywhere 中预定义了一些域，如货币数据类型。也称作用户定义数据类型。

另请参见：“数据类型”一节第 496 页

## 预订

MobiLink 同步中发布与 MobiLink 用户之间的客户端数据库中的一个链接，它使发布所描述的数据能够得到同步。

SQL Remote 复制中发布与远程用户之间的一种链接，它使用户能够与统一数据库交换该发布上的更新。

另请参见：

- [“发布”一节第 484 页](#)
- [“MobiLink 用户”一节第 493 页](#)

## 元数据

数据的数据。元数据描述其它数据的性质和内容。

另请参见：[“模式”一节第 493 页](#)

## 原子事务

保证成功完成或保证根本不予完成的事务。如果错误使原子事务的一部分无法完成，则将回退事务以防止数据库处于不一致的状态。

## REMOTE DBA 特权

在 SQL Remote 中，消息代理 (dbremote) 所需的权限级别。MobiLink 中 SQL Anywhere 同步客户端 (dbmlsync) 所需的权限级别。当消息代理或同步客户端作为具有该权限的用户建立连接时，它将具有完全的 DBA 访问权。如果不是通过消息代理或同步客户端进行连接，则该用户 ID 将不具有附加权限。

另请参见：[“DBA 权限”一节第 484 页](#)

## 远程 ID

SQL Anywhere 和 UltraLite 数据库中一种由 MobiLink 使用的唯一标识符。远程 ID 初始情况下设置为 NULL，在数据库第一次同步期间将设置为 GUID。

## 远程数据库

MobiLink 或 SQL Remote 中一种与统一数据库交换数据的数据库。远程数据库可以共享统一数据库中的全部或部分数据。

另请参见：

- [“同步”一节第 499 页](#)
- [“统一数据库”一节第 499 页](#)

---

## 约束

对特定数据库对象（如表或列）中所包含值的限制。例如，列可以具有唯一性约束，该约束要求该列中的所有值互不相同。表可以具有外键约束，该约束指定该表中的信息与某个其它表中数据的关系。

另请参见：

- [“检查约束”一节第 489 页](#)
- [“外键约束”一节第 501 页](#)
- [“主键约束”一节第 508 页](#)
- [“唯一约束”一节第 502 页](#)

## 运营公司

一种 MobiLink 对象，存储在 MobiLink 系统表或通告程序属性文件中，包含有关供服务器启动的同步使用的公共运营公司的信息。

另请参见：[“服务器启动的同步”一节第 485 页](#)

## 增量备份

仅包含事务日志的备份，通常在两次完全备份之间使用。

另请参见：[“事务日志”一节第 495 页](#)

## 争用

为获取资源而竞争的行为。例如，就数据库而言，如果有两个或更多用户试图编辑数据库的同一行，就会为获得编辑该行的权利而发生争用。

## 正则表达式

正则表达式是字符、通配符和运算符的序列，用于定义某种模式以在字符串内进行搜索。

## 直方图

直方图是列统计信息最重要的组成部分，是一种表示数据分布的方式。SQL Anywhere 维护直方图以为优化程序提供有关列值分布情况的统计信息。

## 直接行处理

MobiLink 中一种用于将表数据同步到 MobiLink 支持的统一数据库以外的数据源的方法。使用直接行处理时，上载和下载都可以实现。

另请参见：

- [“统一数据库”一节第 499 页](#)
- [“基于 SQL 的同步”一节第 489 页](#)

## 主表

包含外键关系中的主键的表。

## 主键

其值唯一标识表中各行中的一个列或多个列。

另请参见：[“外键”一节第 500 页](#)

## 主键约束

一种对主键列的唯一性约束。一个表只能有一个主键约束。

另请参见：

- [“约束”一节第 507 页](#)
- [“检查约束”一节第 489 页](#)
- [“外键约束”一节第 501 页](#)
- [“唯一约束”一节第 502 页](#)
- [“完整性”一节第 501 页](#)

## 子查询

嵌套在 SELECT、INSERT、UPDATE 或 DELETE 语句或者其它子查询中的 SELECT 语句。

有两种类型的子查询：相关子查询和嵌套子查询。

## 字符串

字符串是以单引号围起的字符序列。

## 字符集

字符集是一组符号，包括字母、数字、空格和其它符号。字符集的一个例子是 ISO-8859-1，又称作 Latin1。

另请参见：

- [“代码页”一节第 483 页](#)
- [“编码”一节第 481 页](#)
- [“归类”一节第 487 页](#)

---

# 索引

## 其它

?

UltraLite 输入参数, 308

// 注释指示符

UltraLite 关于, 283

/\* 注释指示符

UltraLite 关于, 283

.NET

UltraLite 引擎支持, 19

.NET 兼容性

ADO.NET 的 UltraLite 驱动程序, 20

^

UltraLite 位运算符, 317

~

UltraLite 位运算符, 317

@data 选项

UltraLite Interactive SQL [dbisql] 实用程序, 229

&

UltraLite 位运算符, 317

% 运算符

模函数, UltraLite, 380

|

UltraLite 位运算符, 317

10054

UltraLite 同步流系统错误, 153

128 位高度加密

UltraLite 用法, 35

UltraLite 连接参数, 214

130 错误

UltraLite 模式升级的 SQL 代码, 58

-a 选项

UltraLite HotSync 管道安装程序 [ulcond11] 实用程序, 243

UltraLite 初始化数据库 [ulinit] 实用程序, 249

UltraLite 装载 XML 到数据库 [ulload] 实用程序, 252

-b 选项

UltraLite 卸载数据到 XML [ulunload] 实用程序, 260

UltraLite 卸载旧数据库 [ulunloadold] 实用程序, 263

-creator-ID

UltraLite HotSync 管道安装程序 [ulcond11] 实用程序, 243

-c 选项

UltraLite HotSync 管道安装程序 [ulcond11] 实用程序, 243

UltraLite Interactive SQL [dbisql] 实用程序, 229

UltraLite 信息 [ulinfo] 实用程序, 245

UltraLite 初始化数据库 [ulinit] 实用程序, 249

UltraLite 卸载数据到 XML [ulunload] 实用程序, 260

UltraLite 卸载旧数据库 [ulunloadold] 实用程序, 263

UltraLite 同步 [ulsync] 实用程序, 255

UltraLite 数据库创建 [ulcreate] 实用程序, 235

UltraLite 消除数据库 [ulerase] 实用程序, 242

UltraLite 装载 XML 到数据库 [ulload] 实用程序, 252

UltraLite 校验数据库实用程序 (ulvalid) 实用程序, 265

-d1 选项

UltraLite Interactive SQL [dbisql] 实用程序, 229

-d 选项

UltraLite HotSync 管道安装程序 [ulcond11] 实用程序, 243

UltraLite Interactive SQL [dbisql] 实用程序, 229

UltraLite SQL 预处理器 [sqlpp] 实用程序, 232

UltraLite 初始化数据库 [ulinit] 实用程序, 249

UltraLite 卸载数据到 XML [ulunload] 实用程序, 260

UltraLite 装载 XML 到数据库 [ulload] 实用程序, 252

-ek 选项

UltraLite 消除数据库 [ulerase] 实用程序, 242

-ep 选项

UltraLite 消除数据库 [ulerase] 实用程序, 242

-e 选项

UltraLite SQL 预处理器 [sqlpp] 实用程序, 232

UltraLite 初始化数据库 [ulinit] 实用程序, 249

UltraLite 卸载数据到 XML [ulunload] 实用程序, 260

UltraLite 校验数据库实用程序 (ulvalid) 实用程序, 265

-f 选项

UltraLite Interactive SQL [dbisql] 实用程序, 229

UltraLite 卸载数据到 XML [ulunload] 实用程序, 260

- UltraLite 卸载旧数据库 [ulunloadold] 实用程序, 263
- UltraLite 装载 XML 到数据库 [ulload] 实用程序, 252
- g 选项
  - UltraLite SQL 预处理器 [sqlpp] 实用程序, 232
  - UltraLite 信息 [ulinfo] 实用程序, 245
  - UltraLite 数据库创建 [ulcreate] 实用程序, 235
  - UltraLite 装载 XML 到数据库 [ulload] 实用程序, 252
- h 选项
  - UltraLite SQL 预处理器 [sqlpp] 实用程序, 232
- I 选项
  - UltraLite 初始化数据库 [ulinit] 实用程序, 249
  - UltraLite 装载 XML 到数据库 [ulload] 实用程序, 252
- k 选项
  - UltraLite SQL 预处理器 [sqlpp] 实用程序, 232
- l 选项
  - UltraLite 初始化数据库 [ulinit] 实用程序, 249
  - UltraLite 数据库创建 [ulcreate] 实用程序, 235
- nogui 选项
  - UltraLite Interactive SQL [dbisql] 实用程序, 229
- n 选项
  - UltraLite HotSync 管道安装程序 [ulcond11] 实用程序, 243
  - UltraLite SQL 预处理器 [sqlpp] 实用程序, 232
  - UltraLite 初始化数据库 [ulinit] 实用程序, 249
  - UltraLite 卸载数据到 XML [ulunload] 实用程序, 260
  - UltraLite 装载 XML 到数据库 [ulload] 实用程序, 252
- oa 选项
  - UltraLite 信息 [ulinfo] 实用程序, 245
  - UltraLite 卸载数据到 XML [ulunload] 实用程序, 260
  - UltraLite 同步 [ulsync] 实用程序, 255
  - UltraLite 装载 XML 到数据库 [ulload] 实用程序, 252
  - UltraLite 校验数据库实用程序 (ulvalid) 实用程序, 265
- ol 选项
  - UltraLite 数据库创建 [ulcreate] 实用程序, 235
  - UltraLite 装载 XML 到数据库 [ulload] 实用程序, 252
- onerror 选项
  - UltraLite Interactive SQL [dbisql] 实用程序, 229
- UltraLite 装载 XML 到数据库 [ulload] 实用程序, 252
- or 选项
  - UltraLite 信息 [ulinfo] 实用程序, 245
  - UltraLite 卸载数据到 XML [ulunload] 实用程序, 260
  - UltraLite 同步 [ulsync] 实用程序, 255
  - UltraLite 装载 XML 到数据库 [ulload] 实用程序, 252
  - UltraLite 校验数据库实用程序 (ulvalid) 实用程序, 265
- output-file
  - UltraLite 卸载数据到 XML [ulunload] 实用程序, 260
- ou 选项
  - UltraLite 信息 [ulinfo] 实用程序, 245
  - UltraLite 卸载数据到 XML [ulunload] 实用程序, 260
  - UltraLite 同步 [ulsync] 实用程序, 255
  - UltraLite 装载 XML 到数据库 [ulload] 实用程序, 252
  - UltraLite 校验数据库实用程序 (ulvalid) 实用程序, 265
- o 选项
  - ulcreate 创建参数, 30
  - UltraLite SQL 预处理器 [sqlpp] 实用程序, 232
  - UltraLite 初始化数据库 [ulinit] 实用程序, 249
  - UltraLite 数据库创建 [ulcreate] 实用程序, 235
  - UltraLite 装载 XML 到数据库 [ulload] 实用程序, 252
- p 选项
  - UltraLite 初始化数据库 [ulinit] 实用程序, 249
  - UltraLite 同步 [ulsync] 实用程序, 255
  - UltraLite 数据库创建 [ulcreate] 实用程序, 235
  - UltraLite 装载 XML 到数据库 [ulload] 实用程序, 252
- qq 选项
  - UltraLite HotSync 管道安装程序 [ulcond11] 实用程序, 243
- q 选项
  - UltraLite HotSync 管道安装程序 [ulcond11] 实用程序, 243
  - UltraLite Interactive SQL [dbisql] 实用程序, 229
  - UltraLite SQL 预处理器 [sqlpp] 实用程序, 232
  - UltraLite 信息 [ulinfo] 实用程序, 245
  - UltraLite 初始化数据库 [ulinit] 实用程序, 249



- UltraLite 卸载数据到 XML [ulunload] 实用程序, 260
- UltraLite 卸载旧数据库 [ulunloadold] 实用程序, 263
- UltraLite 同步 [ulsync] 实用程序, 255
- UltraLite 数据库创建 [ulcreate] 实用程序, 235
- UltraLite 消除数据库 [ulerase] 实用程序, 242
- UltraLite 装载 XML 到数据库 [ulload] 实用程序, 252
- UltraLite 校验数据库实用程序 (ulvalid) 实用程序, 265
- rc 选项
  - UltraLite 信息 [ulinfo] 实用程序, 245
- r 选项
  - UltraLite SQL 预处理器 [sqlpp] 实用程序, 232
  - UltraLite 信息 [ulinfo] 实用程序, 245
  - UltraLite 同步 [ulsync] 实用程序, 255
- SQL 命令
  - UltraLite Interactive SQL [dbisql] 实用程序, 229
- s 选项
  - UltraLite SQL 预处理器 [sqlpp] 实用程序, 232
  - UltraLite 初始化数据库 [ulinit] 实用程序, 249
  - UltraLite 卸载数据到 XML [ulunload] 实用程序, 260
  - UltraLite 同步 [ulsync] 实用程序, 255
  - UltraLite 装载 XML 到数据库 [ulload] 实用程序, 252
- t 选项
  - UltraLite 初始化数据库 [ulinit] 实用程序, 249
  - UltraLite 卸载数据到 XML [ulunload] 实用程序, 260
  - UltraLite 数据库创建 [ulcreate] 实用程序, 235
  - UltraLite 装载 XML 到数据库 [ulload] 实用程序, 252
- ul 选项
  - UltraLite Interactive SQL [dbisql] 实用程序, 229
- u 选项
  - UltraLite HotSync 管道安装程序 [ulcond11] 实用程序, 243
  - UltraLite SQL 预处理器 [sqlpp] 实用程序, 232
- version
  - Interactive SQL [dbisql] 实用程序, 229
- v 选项
  - UltraLite 信息 [ulinfo] 实用程序, 245
  - UltraLite 卸载数据到 XML [ulunload] 实用程序, 260
- UltraLite 卸载旧数据库 [ulunloadold] 实用程序, 263
- UltraLite 同步 [ulsync] 实用程序, 255
- UltraLite 数据库创建 [ulcreate] 实用程序, 235
- UltraLite 消除数据库 [ulerase] 实用程序, 242
- UltraLite 装载 XML 到数据库 [ulload] 实用程序, 252
- UltraLite 校验数据库实用程序 (ulvalid) 实用程序, 265
- w 选项
  - UltraLite SQL 预处理器 [sqlpp] 实用程序, 232
  - UltraLite 初始化数据库 [ulinit] 实用程序, 249
- xml-file
  - UltraLite 卸载旧数据库 [ulunloadold] 实用程序, 263
- x 选项
  - UltraLite Interactive SQL [dbisql] 实用程序, 229
  - UltraLite SQL 预处理器 [sqlpp] 实用程序, 232
  - UltraLite 卸载数据到 XML [ulunload] 实用程序, 260
- y 选项
  - UltraLite 卸载数据到 XML [ulunload] 实用程序, 260
  - UltraLite 卸载旧数据库 [ulunloadold] 实用程序, 263
  - UltraLite 数据库创建 [ulcreate] 实用程序, 235
  - UltraLite 装载 XML 到数据库 [ulload] 实用程序, 252
- z 选项
  - UltraLite SQL 预处理器 [sqlpp] 实用程序, 232
  - UltraLite 数据库创建 [ulcreate] 实用程序, 235
- 注释指示符
  - UltraLite 关于, 283

## A

- ABS 函数
  - UltraLite 语法, 330
- ACOS 函数
  - UltraLite 语法, 330
- ActiveSync
  - UltraLite 部署提供程序文件, 55
  - 为 UltraLite 客户端注册应用程序, 57
  - 部署 MobiLink UltraLite 应用程序, 135
- ADO.NET
  - UltraLite 驱动程序, 20
- AES 加密算法
  - UltraLite fips 创建参数, 176

- UltraLite fips 用法, 35
- UltraLite 用法, 35
- UltraLite 部署步骤, 50
- AllowDownloadDupRows
  - UltraLite 同步参数, 138
- allsync 表
  - UltraLite 同步表, 124
  - UltraLite 概述, 63
- ALL 条件
  - UltraLite SQL, 312
- ALTER DATABASE SCHEMA FROM FILE 语句
  - UltraLite 模式更改影响, 9
  - UltraLite 语法, 424
  - 用法, 58
- ALTER PUBLICATION 语句
  - UltraLite 语法, 425
- ALTER SYNCHRONIZATION PROFILE 语句
  - UltraLite 语法, 426
- ALTER TABLE 语句
  - UltraLite Interactive SQL 示例, 66
  - UltraLite 语法, 428
- AND
  - UltraLite 位运算符, 317
  - UltraLite 逻辑运算符, 311
- ANY 条件
  - UltraLite SQL, 313
- API
  - UltraLite 选项, 20
- ApplyFile 方法
  - UltraLite 替换模式升级, 58
- ARGN 函数
  - UltraLite 语法, 331
- ASCII
  - UltraLite 排序, 32
  - UltraLite 语法, 332
- ASCII 文件
  - UltraLite 导入, 474
- ASIN 函数
  - UltraLite 语法, 332
- ATAN2 函数
  - UltraLite 语法, 334
- ATAN 函数
  - UltraLite 语法, 333
- AUTOINCREMENT
  - UltraLite 语法, 440
- AVG 函数
  - UltraLite 语法, 335
- 安全
  - UltraLite, 35
  - UltraLite AES FIPS 数据库部署, 50
  - UltraLite FIPS 加密, 176
  - UltraLite 启用 TLS 的同步, 51
  - UltraLite 概述, 35
- 安全性
  - UltraLite 用户验证, 44
- 安装
  - UltraLite 在设备上, 47
  - UltraLite 故障排除, 471
- B**
- BETWEEN 条件
  - UltraLite SQL, 313
- BIGINT 数据类型
  - UltraLite, 290
- BINARY 数据类型
  - UltraLite, 290
  - UltraLite 最大大小, 6
- BYTE\_LENGTH 函数
  - UltraLite 语法, 335
- BYTE\_SUBSTR 函数
  - UltraLite 语法, 336
- 版本
  - UltraLite 同步参数, 159
  - UltraLite 故障排除实用程序, 475
- 帮助
  - 技术支持, xvi
- 包
  - 术语定义, 481
- 保留字
  - UltraLite SQL, 280
- 备份 (见 备份)
  - Palm 上的 UltraLite 数据库, 238
  - UltraLite 事务概述, 14
  - UltraLite 内部机制, 13
  - Windows Mobile 上的 UltraLite 数据库, 41
- 备份和数据恢复
  - UltraLite 关于, 13
- 被引用对象
  - 术语定义, 481
- 比较
  - UltraLite 和 SQL Anywhere 数据库, 2
- 比较 UltraLite 和 SQL Anywhere
  - 关于, 2
- 比较运算符

- UltraLite SQL, 310
- UltraLite 的动态 SQL 语法, 310
- 编程接口
  - UltraLite 支持的, 20
- 编辑
  - UltraLite 表方法, 67
- 编码
  - UltraLite utf8\_encoding 创建参数, 193
  - UltraLite utf8\_encoding 用法, 33
  - 术语定义, 481
- 变更
  - UltraLite ALTER PUBLICATION 语句, 425
  - UltraLite ALTER TABLE 语句, 428
  - UltraLite 列方法, 65, 66
  - UltraLite 表方法, 66
- 变更 UltraLite 列定义
  - 关于, 65
- 变量
  - UltraLite SQL, 319
- 标识符
  - UltraLite SQL, 281
  - 术语定义, 481
- 表
  - UltraLite allsync, 124
  - UltraLite ALTER TABLE 语句, 428
  - UltraLite CREATE TABLE 语句, 440
  - UltraLite INSERT 语句, 451
  - UltraLite nosync, 124
  - UltraLite TRUNCATE TABLE 语句, 464
  - UltraLite 中间查询结果存储, 10
  - UltraLite 临时表用法, 320
  - UltraLite 使用, 62
  - UltraLite 使用发布控制同步, 124
  - UltraLite 创建方法, 62
  - UltraLite 删除方法, 66
  - UltraLite 发布, 124
  - UltraLite 变更方法, 66
  - UltraLite 复制方法, 68, 69
  - UltraLite 大小限制, 62
  - UltraLite 批量装载, 452
  - UltraLite 浏览方法, 67
  - UltraLite 编辑方法, 67
  - UltraLite 行发布, 76
  - UltraLite 表过滤方法, 68
  - UltraLite 限制, 6
  - UltraLite 顺序, 126
- 表表达式
  - UltraLite SQL 的子查询, 307
- 表达式
  - UltraLite IF 表达式, 305
  - UltraLite SQL, 303
  - UltraLite SQL 的子查询, 307
  - UltraLite SQL 运算符优先级, 317
  - UltraLite 列名称, 304
  - UltraLite 常量, 304
  - UltraLite 的 CASE 表达式, 305
  - UltraLite 的集合, 306
  - UltraLite 输入参数, 308
- 表达式中的常量
  - UltraLite 关于, 304
- 表达式中的列名
  - UltraLite 关于, 304
- 表大小
  - UltraLite 限制, 6
  - 行数, 6
- 表扫描
  - UltraLite 按主键排序结果, 108
- 表所有者
  - UltraLite, 281
- 表约束
  - UltraLite CREATE TABLE 语句, 440
  - UltraLite 添加、删除或修改, 428
- 别名
  - UltraLite 列, 457
  - UltraLite 等效项, 292
- 并发
  - UltraLite 同步, 11
  - UltraLite 问题, 11
  - 术语定义, 481
- 并发访问
  - UltraLite 引擎, 19
- 不明确的字符串到日期的转换
  - UltraLite, 179
- 部署
  - HotSync 管道文件, 132
  - UltraLite ActiveSync 提供程序文件, 55
  - UltraLite 到设备, 47
  - UltraLite 引擎故障排除, 477
  - UltraLite 模式升级, 58
  - 使用 ActiveSync 的应用程序（为 UltraLite 客户端）, 135
  - 启用了 FIPS 的 UltraLite 应用程序, 176

**C**

- CACHE\_SIZE 连接参数
  - UltraLite 语法, 208
  - 早期版本的 Palm OS 的 UltraLite 优化, 112
- CASE 表达式
  - UltraLite NULLIF 函数, 384
  - UltraLite SQL 语法, 305
- CAST 函数
  - UltraLite 语法, 337
- CE\_FILE 连接参数
  - UltraLite 语法, 209
- CEILING 函数
  - UltraLite 语法, 338
- Certicom
  - UltraLite 加密模块, 35
  - UltraLite 启用 TLS 的同步, 51
  - 用于 Palm 和 Windows Mobile 的 UltraLite 模块, 50
- CHAR\_LENGTH 函数
  - UltraLite 语法, 340
- char\_set 属性
  - UltraLite 说明, 195
- CHARINDEX 函数
  - UltraLite 语法, 339
- CHAR 函数
  - UltraLite 语法, 339
- CHAR 数据类型
  - UltraLite, 290
- CHECK CONSTRAINTS 子句
  - UltraLite LOAD TABLE 语句, 453
- CheckpointStore
  - UltraLite 同步参数, 138
- CHECKPOINT 语句
  - UltraLite 语法, 431
- checksum\_level 创建参数
  - UltraLite 说明, 168
- checksum\_level 属性
  - UltraLite 说明, 195
- CHECK 约束
  - 术语定义, 489
  - 重定向器
    - 术语定义, 482
  - 重命名
    - UltraLite 表, 428
    - 升级过程中的 UltraLite 数据库对象, 58
- COALESCE 函数
  - UltraLite 语法, 341
- CodeWarrior
  - UltraLite 构建 CustDB 应用程序, 89
- COL\_LENGTH 函数
  - 语法, 329
- COL\_NAME 函数
  - 语法, 329
- commit\_flush\_count 属性
  - UltraLite 说明, 195
- commit\_flush\_count 数据库选项
  - UltraLite, 202
- commit\_flush\_timeout 属性
  - UltraLite 说明, 195
- commit\_flush\_timeout 数据库选项
  - UltraLite, 203
- COMMIT\_FLUSH 连接参数
  - UltraLite 语法, 210
- COMMIT 语句
  - UltraLite 语法, 432
- COMPUTES 子句
  - UltraLite LOAD TABLE 语句, 453
- conn\_count 属性
  - UltraLite 说明, 195
- CONVERT 函数
  - UltraLite 语法, 342
- CON 连接参数
  - UltraLite 语法, 211
- COS 函数
  - UltraLite 语法, 344
- COT 函数
  - UltraLite 语法, 345
- count 操作
  - UltraLite 执行计划, 321
- COUNT 函数
  - UltraLite 语法, 345
- CPU
  - UltraLite 限制, 6
- CREATE INDEX 语句
  - UltraLite Interactive SQL 示例, 73
  - UltraLite 语法, 433
  - UNIQUE 参数, 433
- CREATE PUBLICATION 语句
  - UltraLite Interactive SQL 子集示例, 77
  - UltraLite Interactive SQL 整个表示例, 76
  - UltraLite Interactive SQL 示例, 76
  - UltraLite 用法, 125
  - UltraLite 语法, 435

- CREATE SYNCHRONIZATION PROFILE 语句
  - UltraLite 语法, 436
- CREATE TABLE 语句
  - UltraLite Interactive SQL 示例, 63
  - UltraLite 语法, 440
- Crossfire
  - UltraLite 支持, 20
- CROSS JOIN 子句
  - UltraLite 语法, 449
- CSV 文件
  - UltraLite 导入, 474
- 错误回调
  - UltraLite 模式升级错误, 59
- CURRENT DATE
  - UltraLite 特殊值, 286
- CURRENT TIME
  - UltraLite 特殊值, 286
- CURRENT TIMESTAMP
  - UltraLite 特殊值, 2, 287
- CustDB
  - UltraLite 启动, 91
  - UltraLite 应用程序自述文件, 87
  - UltraLite 数据库同步, 94
  - UltraLite 示例, 85
  - UltraLite 示例文件位置, 87
  - UltraLite 限制的实例, 85
  - 构建应用程序的 UltraLite, 89
  - 运行多个实例的 UltraLite, 85
  - 运行应用程序的 UltraLite, 89
- custdb.sql
  - 调用同步脚本, 96
- custdb.udb
  - UltraLite 位置, 87
  - 位置, 87
- C 语言编程
  - UltraLite 支持, 20
- 参考数据库
  - UltraLite 的选项, 27
  - 为 UltraLite 创建, 27
  - 术语定义, 482
- 参数
  - UltraLite SQL 输入, 308
  - UltraLite 数据库创建, 165
  - UltraLite 连接列表, 38
  - UltraLite 连接概述, 42
- 参照完整性
  - UltraLite 数据库, 2
- UltraLite 索引, 71
- UltraLite 表顺序, 126
- 术语定义, 482
- 操作系统
  - UltraLite Windows 支持的平台, 20
- 策略
  - 术语定义, 482
- 插件
  - UltraLite 故障排除, 471
- 插件模块
  - 术语定义, 482
- 插入
  - UltraLite INSERT 语句, 451
  - UltraLite 批量插入行, 452
  - 使用 LOAD TABLE 语句插入 UltraLite 数据, 452
- 查看
  - UltraLite 执行工厂, 320
  - UltraLite 表方法, 67
- 查看 UltraLite 数据库设置
  - 关于, 199
- 查询
  - UltraLite 优化, 458
  - UltraLite 按主键排序结果, 108
  - UltraLite 无法预测结果集故障排除, 476
  - 术语定义, 482
- 查询优化
  - (参见 优化程序)
  - UltraLite SQL, 320
- 查询优化程序
  - (参见 优化程序)
  - UltraLite, 320
- 查找详细信息并请求技术协助
  - 技术支持, xvii
- 常量
  - UltraLite SQL 语法, 304
- 撤消
  - UltraLite 事务, 456
- 持久内存
  - UltraLite 数据库存储, 40
- 池
  - UltraLite 未使用的全局 ID, 119
- 冲突解决
  - 术语定义, 482
- 抽取
  - 术语定义, 482
- 初始化

- 使用 ulinit 初始化 UltraLite 数据库, 249
- 初始化数据库
  - Sybase Central UltraLite 插件, 26
- 触发器
  - UltraLite 限制, 2
  - 术语定义, 483
- 传输规则
  - 术语定义, 483
- 传输文件
  - 使用 MLFileTransfer 传输 UltraLite 文件, 129
- 传送层安全
  - (参见 TLS)
- 窗口 (OLAP)
  - 术语定义, 483
- 创建
  - UltraLite CREATE PUBLICATION 语句, 435
  - UltraLite CREATE TABLE 语句, 440
  - UltraLite 发布, 124
  - UltraLite 发布表, 75
  - UltraLite 数据库方法概述, 25
  - UltraLite 用户, 79
  - UltraLite 索引, 73
  - UltraLite 表方法, 62
  - UltraLite 远程数据库, 118
  - 从 MobiLink 同步模型创建 UltraLite 数据库, 27
  - 从 XML 创建 UltraLite 数据库, 28
  - 从命令提示符创建 UltraLite 数据库, 26
  - 使用 ulcreate 创建 UltraLite 数据库, 235
  - 使用 ulinit 创建 UltraLite 数据库, 249
  - 用于 UltraLite 的参考数据库, 27
- 创建 UltraLite 表
  - 关于, 62
- 创建 UltraLite 数据库
  - 关于, 25
- 创建表向导
  - UltraLite 用法, 63
- 创建参数 (UltraLite)
  - checksum\_level, 168
  - date\_format, 171
  - date\_order, 174
  - fips 创建参数, 176
  - max\_hash\_size, 177
  - nearest\_century, 179
  - obfuscate, 181
  - page\_size, 182
  - time\_format, 187
  - timestamp\_format, 189
  - timestamp\_increment, 192
  - utf8\_encoding, 193
  - 关于, 165
  - 大小写, 167
  - 小数位数, 185
  - 归类, 170
  - 精度, 184
- 创建发布
  - UltraLite 数据库与 MobiLink 应用程序, 124
- 创建发布向导
  - UltraLite 用法, 75
  - UltraLite 行发布, 76
- 创建数据库
  - Sybase Central UltraLite 插件, 26
  - UltraLite 创建参数, 165
  - UltraLite 远程, 118
- 创建数据库向导
  - UltraLite 用法, 26
- 创建索引向导
  - UltraLite 使用, 72
- 创建用户向导
  - UltraLite 用法, 79
- 创建远程数据库
  - UltraLite 客户端, 118
- 创建者 ID
  - 术语定义, 483
- 从系统故障恢复
  - UltraLite 内部机制, 13
- 存储
  - UltraLite PALM\_ALLOW\_BACKUP, 220
  - UltraLite 保留大小, 224
  - UltraLite 数据库, 9
  - UltraLite 限制, 6
- 存储过程
  - UltraLite 限制, 2
  - 术语定义, 483
- 错误
  - UltraLite SQLE\_DATABASE\_ERROR, 472
  - UltraLite SQLE\_DEVICE\_ERROR, 472
  - UltraLite SQLE\_MEMORY\_ERROR, 472
  - UltraLite 客户端错误 -764, 477
  - 提供反馈, xvi
- 错误代码
  - UltraLite 卸载数据到 XML [ulunload] 实用程序, 228
  - UltraLite 实用程序, 228
  - UltraLite 数据库创建 [ulcreate] 实用程序, 228

---

UltraLite 数据库同步 [ulsync] 实用程序, 228  
UltraLite 装载 XML 到数据库 [ulload] 实用程序,  
228

## D

DATALENGTH 函数

UltraLite 语法, 346

date\_format 创建参数

UltraLite 说明, 171

date\_format 属性

UltraLite 说明, 195

date\_order 创建参数

UltraLite 说明, 174

date\_order 属性

UltraLite 说明, 195

DATEADD 函数

UltraLite 语法, 348

DATEDIFF 函数

UltraLite 语法, 349

DATEFORMAT 函数

UltraLite 语法, 350

DATENAME 函数

UltraLite 语法, 351

DATEPART 函数

UltraLite 语法, 351

DATETIME 函数

UltraLite 语法, 352

DATE 函数

UltraLite 语法, 347

DATE 数据类型

UltraLite, 290

DAYNAME 函数

UltraLite 语法, 353

DAYS 函数

UltraLite 语法, 354

DAY 函数

UltraLite 语法, 353

DB\_PROPERTY 函数

UltraLite 语法, 355

DBA 权限

术语定义, 484

DBF 连接参数

UltraLite 语法, 212

dbhsync11.dll

UltraLite 中的 HotSync 管道, 132

dbisql.exe

安装前关闭, 471

dbisql 实用程序

UltraLite 数据导入故障排除, 474

UltraLite 语法, 229

UltraLite 退出代码, 231

DBKEY 连接参数

UltraLite 语法, 214

dblgen11.dll

UltraLite 中的 ActiveSync 管道部署, 55

UltraLite 中的 HotSync 管道部署, 132

dbmlhttp11.dll

部署 UltraLite 应用程序, 132

dbmlhttps11.dll

部署 UltraLite 应用程序, 132

dbmlsock11.dll

部署 UltraLite 应用程序, 132

dbmltls11.dll

部署 UltraLite 应用程序, 132

DBMS

术语定义, 497

DBN 连接参数

UltraLite 语法, 215

dbser11.dll

部署 UltraLite 应用程序, 132

dbspaces

术语定义, 484

DCX

关于, xii

DDL

术语定义, 496

DDL 语句

UltraLite 模式更改, 9

DECIMAL 数据类型

UltraLite, 290

DEFAULTS 子句

UltraLite LOAD TABLE 语句, 453

DEFAULT TIMESTAMP 列

UltraLite 语法, 440

DEGREES 函数

语法, 356

DELETE 语句

UltraLite 动态 SQL 语法, 444

DELIMITED BY 子句

UltraLite LOAD TABLE 语句, 453

DIFFERENCE 函数

UltraLite 语法, 356

DisableConcurrency

UltraLite 同步参数, 138

- disable concurrency
  - UltraLite 同步参数概述, 11
- distinct 操作
  - UltraLite 执行计划, 321
- DISTINCT 关键字
  - UltraLite SQL, 457
- DML
  - 术语定义, 496
- DocCommentXchange (DCX)
  - 关于, xii
- DOUBLE 数据类型
  - UltraLite, 290
- download\_only 同步参数
  - UltraLite 参考, 142
- DOW 函数
  - UltraLite 语法, 357
- DROP INDEX 语句
  - UltraLite Interactive SQL 示例, 74
  - UltraLite 语法, 445
- DROP PUBLICATION 语句
  - UltraLite Interactive SQL 示例, 77
  - UltraLite 语法, 446
- DROP SYNCHRONIZATION PROFILE 语句
  - UltraLite 语法, 447
- DROP TABLE 语句
  - UltraLite Interactive SQL 示例, 67
  - UltraLite 语法, 448
- dummy 操作
  - UltraLite 执行计划, 321
- 大小写创建参数
  - UltraLite 说明, 167
- 大小写属性
  - UltraLite 说明, 195
- 大小写数据库属性
  - UltraLite 数据库创建 [ulcreate] 实用程序, 235
- 大写字符
  - UltraLite UPPER 函数, 414
- 大写字符串
  - UltraLite UCASE 函数, 413
  - UltraLite UPPER 函数, 414
- 代理 ID
  - 术语定义, 483
- 代理表
  - 术语定义, 483
- 代码点
  - UltraLite, 32
- 代码页
  - 术语定义, 483
- 当前日期函数
  - UltraLite TODAY 函数, 410
- 当前行
  - UltraLite 并发, 11
- 导出
  - 使用 unload 导出 UltraLite 数据库, 260
- 导出工具
  - UltraLite unload 实用程序, 260
- 导入数据
  - UltraLite 故障排除, 474
- 调整性能
  - UltraLite max\_hash\_size, 177
- 动态 SQL
  - UltraLite 位运算符, 317
  - UltraLite 字符串运算符, 317
  - UltraLite 算术运算符, 316
  - UltraLite 运算符优先级, 317
  - UltraLite 逻辑运算符, 311
  - 术语定义, 484
- 逗号分隔的列表
  - UltraLite LIST 函数语法, 371
- 读取
  - UltraLite, 15
  - UltraLite 表行, 15
- 读取 UltraLite 访问计划
  - 关于, 321
- 读取行
  - UltraLite 并发, 15
- 端口号
  - UltraLite ULSynchronize 参数, 155
- 对象树
  - 术语定义, 484
- 多表连接
  - UltraLite 数据库, 2
- 多个设备
  - UltraLite 连接参数, 40
- 多个数据库
  - UltraLite 最大, 11
- 多进程访问
  - UltraLite 引擎, 19
- 多线程
  - UltraLite 应用程序, 11
- 多线程应用程序
  - UltraLite 关于, 19



## E

### EBF

术语定义, 484

### ELSE

UltraLite IF 表达式, 305

UltraLite 的 CASE 表达式, 305

### eMbedded Visual C++

UltraLite CustDB 示例和自述文件, 89

### ENCODING 子句

UltraLite LOAD TABLE 语句, 453

### END

UltraLite 的 CASE 表达式, 305

### ENDIF

UltraLite IF 表达式, 305

### ERRORLEVEL 环境变量

UltraLite 的 Interactive SQL 返回代码, 231

### ER 图选项卡

UltraLite 关于, 69

### ESCAPES 子句

UltraLite LOAD TABLE 语句, 453

### ESQL (非 UltraLite SQL) (非 嵌入式 SQL)

### EXISTS 条件

UltraLite SQL, 314

### EXPLANATION 函数

UltraLite 语法, 359

### EXP 函数

UltraLite 语法, 359

### 二进制

UltraLite 排序, 32

## F

### 非可重复的读取

UltraLite 关于, 15

### FILE

术语定义, 485

### FILE 消息类型

术语定义, 485

### filter 操作

UltraLite 执行计划, 321

### FIPS

UltraLite fips 属性用法, 35

UltraLite 加密数据库部署, 50

UltraLite 启用 TLS 的同步, 51

UltraLite 设置和部署, 176

### FIPS 创建参数

UltraLite 说明, 176

### fips 数据库属性

UltraLite 用法, 35

UltraLite 部署步骤, 50

### fips 网络协议选项

UltraLite 启用 TLS 的同步, 51

### FIRST 子句

UltraLite SELECT 语句, 457

### FLOAT 数据类型

UltraLite, 290

### FLOOR 函数

UltraLite 语法, 360

### FORCE ORDER 子句

UltraLite SELECT 语句, 458

### FORMAT 子句

UltraLite LOAD TABLE 语句, 453

### FOR READ ONLY 子句

UltraLite 直接页扫描使用, 108

### FOR 子句

UltraLite SELECT 语句, 458

### FROM 子句

UltraLite LOAD TABLE 语句, 452

UltraLite SELECT 语句, 457

UltraLite 语法, 449

### 发布

UltraLite ALTER PUBLICATION 语句, 425

UltraLite CREATE PUBLICATION 语句, 435

UltraLite SQL CREATE INDEX 语句, 433

UltraLite SQL DROP INDEX 语句, 445

UltraLite SQL DROP PUBLICATION 语句, 446

UltraLite SQL DROP TABLE 语句, 448

UltraLite sysarticle 系统表, 275

UltraLite syspublication 系统表, 274

UltraLite WHERE 子句用法, 76

UltraLite 使用, 75

UltraLite 删除, 77

UltraLite 发布概述, 124

UltraLite 发布表, 75

UltraLite 同步, 149

UltraLite 同步参数, 149

UltraLite 复制方法, 69

UltraLite 整个表, 124

UltraLite 模式描述, 274

UltraLite 行, 76

UltraLite 行发布, 76

UltraLite 表, 75

UltraLite 设计计划, 123

UltraLite 限制, 249

- 术语定义, 484
- 模式中列出的 UltraLite 表, 275
- 发布创建向导
  - UltraLite 创建发布, 125
- 发布更新
  - 术语定义, 484
- 发布者
  - 术语定义, 485
- 发布整个表
  - UltraLite, 75
- 发送列名
  - UltraLite 同步参数, 151
- 发送下载确认
  - UltraLite 同步参数, 152
- 反馈
  - 报告错误, xvi
  - 提供, xvi
  - 文档, xvi
  - 请求更新, xvi
- 反余弦函数
  - UltraLite ACOS 函数, 330
- 反正切函数
  - UltraLite ATAN 函数, 333
- 反正弦函数
  - UltraLite ASIN 函数, 332
- 返回代码
  - UltraLite 的 Interactive SQL [dbisql] 实用程序, 231
- 范围
  - UltraLite 数据类型, 290
- 非唯一索引
  - UltraLite 特性, 100
  - UltraLite 索引创建, 72
- 分区
  - UltraLite 主键, 119
  - UltraLite 行发布, 76
- 分区大小
  - UltraLite 用尽缺省值, 119
  - UltraLite 缺省值, 替换, 121
  - UltraLite 缺省值, 选择, 119
- 分析树
  - 术语定义, 485
- 服务
  - 术语定义, 485
- 服务器管理请求
  - 术语定义, 485
- 服务器启动的同步

- 术语定义, 485
- 服务器消息存储库
  - 术语定义, 485
- 复制
  - UltraLite 表方法, 68
  - 术语定义, 485
- 复制代理
  - 术语定义, 485
- 复制服务器
  - 术语定义, 486
- 复制频率
  - 术语定义, 486
- 复制数据
  - UltraLite 数据库, 68
- 复制消息
  - 术语定义, 486
- 附加参数
  - UltraLite 同步参数, 138

## G

- GETDATE 函数
  - UltraLite 语法, 361
- GetLastIdentity 方法
  - UltraLite 同步, 120
- getScriptVersion 方法
  - UltraLite 示例, 159
- getStream 方法
  - UltraLite 示例, 154
- getUploadOK 方法
  - UltraLite 示例, 156
- global\_database\_id 属性
  - UltraLite 说明, 195
- global\_database\_id 选项
  - UltraLite, 204
  - UltraLite CREATE TABLE 语句, 440
  - UltraLite, 设置, 119
- GREATER 函数
  - UltraLite 语法, 361
- group-by 操作
  - UltraLite 执行计划, 321
- GROUP BY 子句
  - UltraLite SELECT 语句, 457
- GUID
  - MobiLink 系统中的 UltraLite 客户端, 119
  - NEWID 函数的 UltraLite SQL 语法, 383
  - STRTOUUID 函数的 UltraLite SQL 语法, 404
  - UIDTOSTR 函数的 UltraLite SQL 语法, 416

- 高度加密
    - UltraLite fips 创建参数, 176
    - UltraLite 性能影响, 111
    - UltraLite 用法, 35
    - UltraLite 部署步骤, 50
  - 高速缓存
    - UltraLite 性能, 111
    - UltraLite 最大大小, 6
  - 高速缓存大小
    - UltraLite 用法, 182
    - UltraLite 限制, 6
  - 格式选项 (UltraLite)
    - utf8\_encoding 用法, 33
  - 隔离级别
    - UltraLite 关于, 15
    - UltraLite 脏读, 15
    - UltraLite 隔离级别, 15
    - 术语定义, 486
  - 个人服务器
    - 术语定义, 486
  - 跟踪
    - UltraLite 同步, 116
  - 更新
    - UltraLite 数据库, 12
    - UltraLite 更新行, 467
  - 工作表
    - 术语定义, 486
  - 功能
    - UltraLite 比较列表, 2
  - 公共证书
    - UltraLite 应用程序访问, 256
  - 构建
    - UltraLite CustDB 应用程序, 89
  - 故障
    - UltraLite 模式升级错误, 59
    - UltraLite 防止内存不足的错误, 224
  - 故障排除
    - UltraLite 保持时间戳和时间戳增量, 189
    - UltraLite 启用自动备份, 238
    - UltraLite 校验和故障, 168
    - UltraLite 连接参数传送, 42
  - 故障切换
    - 术语定义, 486
  - 关键连接
    - 术语定义, 496
  - 关键字
    - UltraLite SQL, 280
  - 管道
    - UltraLite HotSync, 53
    - UltraLite HotSync 管道文件, 53
    - 为 CustDB 安装, 243
    - 安装, 243
    - 部署 UltraLite 应用程序, 132
  - 管道文件
    - 部署 HotSync, 132
  - 管理工具
    - UltraLite 实用程序参考, 227
    - UltraLite 故障排除, 471
  - 管理临时表
    - UltraLite 关于, 108
  - 管理实用程序
    - UltraLite 实用程序参考, 227
  - 管理事务
    - UltraLite 提高的吞吐量, 110
  - 管理数据库
    - UltraLite 数据和状态, 11
  - 规范化
    - 术语定义, 487
  - 归类
    - UltraLite CollationName 属性, 195
    - UltraLite 不支持, 250
    - UltraLite 归类创建参数, 170
    - 术语定义, 487
  - 归类创建参数
    - UltraLite 说明, 170
  - 归类属性
    - UltraLite 说明, 195
  - 归类序列
    - UltraLite 关于, 32
    - UltraLite 更改, 32
  - 过程
    - UltraLite 限制, 2
  - 过滤
    - UltraLite 表方法, 68
  - 过滤器
    - UltraLite 文件名, 238
- ## H
- HEXTOINT 函数
    - UltraLite 语法, 362
  - HotSync
    - 部署管道文件, 132
  - HotSync 管道
    - 为 CustDB 安装, 243

- 安装, 243
- HotSync 配置概述
  - UltraLite 客户端, 53
- HotSync 同步
  - UltraLite Palm OS, 53
  - UltraLite 体系结构, 132
  - UltraLite 客户端, 133
  - 配置, 133
- HOURS 函数
  - UltraLite 语法, 364
- HOUR 函数
  - UltraLite 语法, 363
- 回调
  - UltraLite 模式升级错误, 59
- 函数
  - UltraLite 函数按字母顺序排序的列表 A-D, 330
  - UltraLite 函数按字母顺序排序的列表 E-O, 359
  - UltraLite 函数按字母顺序排序的列表 P-Z, 386
  - UltraLite 函数的类型, 324
  - UltraLite 字符串, 327
  - UltraLite 数字, 327
  - UltraLite 数据类型转换 SQL, 324
  - UltraLite 日期和时间, 324
  - UltraLite 杂类, 326
  - UltraLite 简介, 323
  - UltraLite 系统 SQL, 328
  - UltraLite 集合, 324
  - 如果指定参数 NULL 则返回 NULL, 323
- 函数, 集合
  - UltraLite AVG, 335
  - UltraLite COUNT, 345
  - UltraLite LIST, 371
  - UltraLite MAX, 376
  - UltraLite MIN, 376
  - UltraLite SUM, 406
  - 关于, 324
- 函数, 日期和时间
  - UltraLite DATE, 347
  - UltraLite DATEADD, 348
  - UltraLite DATEDIFF, 349
  - UltraLite DATEFORMAT, 350
  - UltraLite DATENAME, 351
  - UltraLite DATEPART, 351
  - UltraLite DATETIME, 352
  - UltraLite DAY, 353
  - UltraLite DAYNAME, 353
  - UltraLite DAYS, 354
  - UltraLite DOW, 357
  - UltraLite GETDATE, 361
  - UltraLite HOUR, 363
  - UltraLite HOURS, 364
  - UltraLite MINUTE, 377
  - UltraLite MINUTES, 378
  - UltraLite MONTH, 381
  - UltraLite MONTHNAME, 381
  - UltraLite MONTHS, 382
  - UltraLite NOW, 384
  - UltraLite QUARTER, 388
  - UltraLite SECOND, 395
  - UltraLite SECONDS, 396
  - UltraLite TODAY, 410
  - UltraLite WEEKS, 417
  - UltraLite YEAR, 418
  - UltraLite YEARS, 419
  - UltraLite YMD, 420
  - UltraLite 关于, 324
- 函数, 数据类型转换
  - UltraLite CAST, 337
  - UltraLite CONVERT, 342
  - UltraLite HEXTOINT, 362
  - UltraLite INTTOHEX, 366
  - UltraLite ISDATE, 367
  - UltraLite ISNULL, 367
  - UltraLite 关于, 324
- 函数, 数字
  - DEGREES, 356
  - UltraLite ABS, 330
  - UltraLite ACOS, 330
  - UltraLite ASIN, 332
  - UltraLite ATAN, 333
  - UltraLite ATAN2, 334
  - UltraLite CEILING, 338
  - UltraLite COS, 344
  - UltraLite COT, 345
  - UltraLite EXP, 359
  - UltraLite FLOOR, 360
  - UltraLite LOG, 373
  - UltraLite LOG10, 373
  - UltraLite MOD, 380
  - UltraLite PI, 387
  - UltraLite POWER, 387
  - UltraLite RADIANS, 389
  - UltraLite REMAINDER, 389
  - UltraLite ROUND, 394

- UltraLite SIGN, 398
- UltraLite SIN, 399
- UltraLite SQRT, 401
- UltraLite TAN, 409
- UltraLite TRUNCNUM, 411
- UltraLite 关于, 327
- 函数, 系统
  - DB\_PROPERTY, 355
  - ML\_GET\_SERVER\_NOTIFICATION, 379
  - SYNC\_PROFILE\_OPTION\_VALUE, 408
  - UltraLite DATALENGTH, 346
- 函数, 杂类
  - UltraLite ARGN, 331
  - UltraLite COALESCE, 341
  - UltraLite EXPLANATION, 359
  - UltraLite GREATER, 361
  - UltraLite IFNULL, 365
  - UltraLite LESSER, 370
  - UltraLite NEWID, 383
  - UltraLite NULLIF, 384
  - UltraLite 关于, 326
- 函数, 杂项
  - UltraLite SHORT\_PLAN, 397
- 函数, 字符串
  - UltraLite ASCII, 332
  - UltraLite BYTE\_LENGTH, 335
  - UltraLite BYTE\_SUBSTR, 336
  - UltraLite CHAR, 339
  - UltraLite CHAR\_LENGTH, 340
  - UltraLite CHARINDEX, 339
  - UltraLite DIFFERENCE, 356
  - UltraLite INSERTSTR, 366
  - UltraLite LCASE, 368
  - UltraLite LEFT, 369
  - UltraLite LENGTH, 370
  - UltraLite LOCATE, 372
  - UltraLite LOWER, 374
  - UltraLite LTRIM, 375
  - UltraLite PATINDEX, 386
  - UltraLite REPEAT, 390
  - UltraLite REPLACE, 391
  - UltraLite REPLICATE, 392
  - UltraLite RIGHT, 393
  - UltraLite RTRIM, 394
  - UltraLite SIMILAR, 398
  - UltraLite SOUNDEX, 400
  - UltraLite SPACE, 401
  - UltraLite STR, 402
  - UltraLite STRING, 403
  - UltraLite STRTOUUID, 404
  - UltraLite STUFF, 404
  - UltraLite SUBSTRING, 405
  - UltraLite TRIM, 410
  - UltraLite UCASE, 413
  - UltraLite UPPER, 414
  - UltraLite UUIDTOSTR, 416
  - UltraLite 关于, 327
- 函数, 系统
  - SUSER\_ID, 407
  - SUSER\_NAME, 408
  - TSEQUAL, 412
  - USER\_ID, 414
  - USER\_NAME, 415
- 函数语法
  - UltraLite 函数按字母顺序排序的列表 A-D, 330
  - UltraLite 函数按字母顺序排序的列表 E-O, 359
  - UltraLite 函数按字母顺序排序的列表 P-Z, 386
- 忽略的行
  - UltraLite 同步参数, 143
- 环境变量
  - UltraLite ULSQLCONNECT, 46
  - UltraLite 用法, 43
  - UltraLite 的 ERRORLEVEL, 231
  - 命令 shell, xv
  - 命令提示符, xv
- 幻像行
  - UltraLite, 15
- 恢复
  - UltraLite, 13
  - UltraLite 事务概述, 14
  - UltraLite 简介, 12
- 回退
  - UltraLite 事务, 456
  - UltraLite 事务概述, 14
  - UltraLite 数据库, 12
- 回退日志
  - 术语定义, 487
- 获取帮助
  - 技术支持, xvi
- I
  - iAnywhere JDBC 驱动程序
    - 术语定义, 487
  - iAnywhere 开发人员社区

- 新闻组, xvii
- ID
  - ml\_remote\_id 选项, 205
  - UltraLite 全局数据库, 204
  - UltraLite 用户, 44
- IFNULL 函数
  - UltraLite 语法, 365
- IF 表达式
  - UltraLite SQL 语法, 305
- ignored\_rows 同步参数
  - UltraLite 参考, 143
- INDEX\_COL 函数
  - 语法, 329
- index-scan 操作
  - UltraLite 执行计划, 321
- InfoMaker
  - 术语定义, 487
- INNER JOIN 子句
  - UltraLite 语法, 449
- INPUT 语句
  - UltraLite 故障排除, 474
- INSERTSTR 函数
  - UltraLite 语法, 366
- INSERT 语句
  - UltraLite Interactive SQL 示例, 69
  - UltraLite 语法, 451
- install-dir
  - 文档用法, xiv
- INTEGER 数据类型
  - UltraLite, 290
- Interactive SQL
  - UltraLite 函数按字母顺序排序的列表 A-D, 330
  - UltraLite 函数按字母顺序排序的列表 E-O, 359
  - UltraLite 函数按字母顺序排序的列表 P-Z, 386
  - UltraLite 命令行, 229
  - UltraLite 数据导入故障排除, 474
  - UltraLite 文本计划, 320
  - UltraLite 显示计划, 320
  - UltraLite 计划操作, 321
  - UltraLite 计划解释, 321
  - 术语定义, 488
- Interactive SQL 实用程序 [dbisql]
  - UltraLite 语法, 229
  - UltraLite 退出代码, 231
- INTTOHEX 函数
  - UltraLite 语法, 366
- IN 条件
  - UltraLite SQL, 314
- IS
  - UltraLite 逻辑运算符, 311
- ISDATE 函数
  - UltraLite 语法, 367
- ISNULL 函数
  - UltraLite 语法, 367
- J**
- JAR 文件
  - 术语定义, 488
- Java 类
  - 术语定义, 488
- jConnect
  - 术语定义, 488
- JDBC
  - 术语定义, 488
- 校验
  - (参见 校验数据库)
  - 带有校验数据库向导的 UltraLite 数据库, 12
  - 带 ulvalid 的 UltraLite 数据库, 265
  - 术语定义, 490
- 校验和
  - 术语定义, 490
- 校验数据库
  - UltraLite 校验数据库实用程序 (ulvalid), 265
- 校验数据库实用程序
  - UltraLite 关于, 265
- join 操作
  - UltraLite 执行计划, 321
- 基表
  - 术语定义, 488
- 基础
  - UltraLite 数据库管理, 11
- 基于 SQL 的同步
  - 术语定义, 489
- 基于会话的同步
  - 术语定义, 488
- 基于脚本的上载
  - 术语定义, 489
- 基于索引的 UltraLite 优化
  - UltraLite 关于, 100
- 基于文件的下载
  - 术语定义, 489
- 机制
  - UltraLite 应用程序和数据库部署, 47
- 集成登录

- 术语定义, 489
- 集合表达式
  - UltraLite SQL 语法, 306
- 集合函数
  - UltraLite 按字母顺序排序的列表, 324
- 级联更新
  - UltraLite 限制, 2
- 级联删除
  - UltraLite 限制, 2
- 技术支持
  - 新闻组, xvii
- 计划
  - UltraLite 同步设计概述, 123
  - UltraLite 操作, 321
  - UltraLite 文本计划, 320
  - UltraLite 查询, 使用, 320
  - UltraLite 查询, 替换, 320
  - UltraLite 查询, 读取, 321
  - UltraLite 游标, 359
  - UltraLite 计划操作, 321
  - UltraLite 计划解释, 321
  - UltraLite 语法, 359
- 计算列
  - UltraLite 限制, 2
- 记录
  - UltraLite 内部机制, 13
- 加密
  - UltraLite fips 创建参数, 176
  - UltraLite fips 属性用法, 35
  - UltraLite fips 用法, 35
  - UltraLite obfuscate 创建参数, 181
  - UltraLite obfuscate 属性用法, 35
  - UltraLite TLS 同步配置, 51
  - UltraLite 加密密钥, 214
  - UltraLite 加密属性, 195
  - UltraLite 同步部署注意事项, 51
  - UltraLite 性能影响, 111
  - UltraLite 数据部署注意事项, 50
  - UltraLite 更改密钥, 176
  - UltraLite 部署步骤, 50
- 加密密钥
  - UltraLite 更改, 176
- 加密属性
  - UltraLite 说明, 195
- 加密算法
  - UltraLite Certicom 模块, 35
- 监控同步
  - UltraLite observer 同步参数, 146
  - UltraLite setObserver 方法, 146
- 监听器
  - 术语定义, 489
- 兼容性
  - UltraLite SQL, 310
- 检查点
  - UltraLite 性能优化, 110
  - 术语定义, 489
- 检查点操作
  - UltraLite CHECKPOINT 语法, 431
- 检查约束
  - UltraLite 限制, 2
- 简单加密
  - UltraLite 性能影响, 111
- 键
  - UltraLite 主要的, 63
  - UltraLite 索引创建, 72
  - UltraLite 索引散列, 177
- 建模
  - 从 MobiLink 建模 UltraLite 数据库, 27
- 将参数装配到连接字符串
  - UltraLite 关于, 43
- 将数据导入到数据库
  - UltraLite ulload 实用程序, 252
- 脚本
  - 术语定义, 489
- 脚本版本
  - UltraLite getScriptVersion , 159
  - UltraLite setScriptVersion 方法, 159
  - UltraLite version 同步参数, 159
  - 术语定义, 490
- 脚本式上载
  - UltraLite CREATE PUBLICATION 语法, 435
- 角色
  - 术语定义, 490
- 角色名
  - UltraLite 外键, 440
  - UltraLite 角色名, 440
  - 术语定义, 490
- 教程
  - UltraLite CustDB 数据库同步, 94
  - UltraLite CustDB 文件, 87
  - UltraLite CustDB 简介, 85
- 截断
  - UltraLite 表, 464
- 结果集

- UltraLite 无法预测更改故障排除, 476
- 结果集排序
  - UltraLite 故障排除, 476
- 解压缩行
  - UltraLite 关于, 62
- 介质故障
  - UltraLite 事务概述, 14
- 仅上载
  - UltraLite 同步参数, 157
- 仅上载同步
  - UltraLite upload\_only 同步参数, 157
  - UltraLite 数据库, 157
- 仅下载
  - UltraLite 同步参数, 142
- 仅下载同步
  - UltraLite download\_only 同步参数, 142
  - UltraLite 同步参数, 142
  - UltraLite 定义概述, 127
- 进程中运行时
  - UltraLite 关于, 19
- 进度计数器
  - UltraLite 偏移不匹配, 116
- 精度创建参数
  - UltraLite 说明, 184
- 精度属性
  - UltraLite 说明, 195
- 镜像日志
  - 术语定义, 490
- 局部临时表
  - 术语定义, 490

## K

- 可重新启动的下载
  - UltraLite keep partial download, 144
  - UltraLite partial download retained, 146
  - UltraLite resume partial download, 150
- keep partial download 同步参数
  - UltraLite 参考, 144
- KEY JOIN 子句
  - UltraLite 语法, 449
- keyset 操作
  - UltraLite 执行计划, 321
- 开发平台
  - UltraLite 支持, 20
- 开发人员社区
  - 新闻组, xvii
- 开销

- UltraLite 考虑保留大小, 224
- 客户端
  - UltraLite MobiLink 客户端, 115
  - UltraLite 嵌入式引擎, 19
- 客户端/服务器
  - 术语定义, 490
- 客户端数据库
  - UltraLite 选项, 138
- 客户端消息存储库
  - 术语定义, 490
- 客户端消息存储库 ID
  - 术语定义, 491
- 空格
  - UltraLite 文件路径的定义, 470
- 空数据库
  - 运行 ulinit 后 UltraLite 填充, 250
- 控制同步
  - UltraLite 发布, 124
- 口令
  - PWD UltraLite 连接参数, 223
  - UltraLite password 同步参数, 147
  - UltraLite 数据库, 44
  - UltraLite 新 mobilink 口令参数, 145
  - UltraLite 更改, 79
  - UltraLite 注意事项, 79
  - UltraLite 添加新的, 44
  - UltraLite 缺省值, 79
  - UltraLite 语义, 44
- 口令连接参数
  - UltraLite 语法, 223
- 库
  - UltraLite 将 uleng 部署到 Windows Mobile, 49
  - UltraLite 选项, 20
  - 启用了 FIPS 的 UltraLite 应用程序, 176
- 快照隔离
  - 术语定义, 491
- 扩展卡
  - UltraLite 写入, 41
- 扩展名
  - UltraLite 不允许, 238

## L

- LCASE 函数
  - UltraLite 语法, 368
- LEFT OUTER JOIN 子句
  - UltraLite 语法, 449
- LEFT 函数



- UltraLite 语法, 369
- LENGTH 函数
  - UltraLite 语法, 370
- LESSER 函数
  - UltraLite 语法, 370
- like-scan 操作
  - UltraLite 执行计划, 321
- LIST 函数
  - UltraLite 语法, 371
- LOAD TABLE 语句
  - UltraLite 语法, 452
- LOCATE 函数
  - UltraLite 语法, 372
- LOG10 函数
  - UltraLite 语法, 373
- LOG 函数
  - UltraLite 语法, 373
- lojoin 操作
  - UltraLite 执行计划, 321
- LONG BINARY 数据类型
  - UltraLite, 290
- LONG VARCHAR 数据类型
  - UltraLite, 290
- LOWER 函数
  - UltraLite 语法, 374
- LTM
  - 术语定义, 492
- LTRIM 函数
  - UltraLite 语法, 375
- 联邦信息处理标准 (FIPS)
- 联合
  - UltraLite 多个选择语句, 466
- 联机手册
  - PDF, xii
- 连接
  - MobiLink UltraLite Stream Type 同步参数, 154
  - UltraLite conn\_count 属性, 195
  - UltraLite FROM 子句语法, 449
  - UltraLite 并发, 11
  - UltraLite 故障排除, 471
  - UltraLite 数据库, 44
  - UltraLite 数据库疑难解答, 43
  - UltraLite 概述, 37
  - 术语定义, 491
- 连接 ID
  - 术语定义, 491
- 连接参数
  - UltraLite, 37
  - UltraLite CACHE\_SIZE, 208
  - UltraLite CE\_FILE, 209
  - UltraLite COMMIT\_FLUSH, 210
  - UltraLite CON, 211
  - UltraLite DBF, 212
  - UltraLite DBKEY, 214
  - UltraLite file\_name, 212
  - UltraLite MIRROR\_FILE, 216
  - UltraLite NT\_FILE, 218
  - UltraLite PALM\_ALLOW\_BACKUP, 220
  - UltraLite PALM\_FILE, 221
  - UltraLite PWD, 223
  - UltraLite RESERVE\_SIZE, 224
  - UltraLite START, 225
  - UltraLite UID, 226
  - UltraLite userid, 226
  - UltraLite 优先级, 43
  - UltraLite 传送故障排除, 42
  - UltraLite 列表, 38
  - UltraLite 口令, 223
  - UltraLite 密钥, 214
  - UltraLite 提供, 39
  - UltraLite 概述, 42
  - UltraLite 的 DBN, 215
  - UltraLite 连接摘要, 42
  - UltraLite 选择, 40
  - 按字母顺序排序的列表 (UltraLite), 207
- 连接到 UltraLite 数据库
  - 关于, 37
- 连接方法
  - UltraLite 关于, 37
- 连接类型
  - 术语定义, 491
- 连接配置
  - 术语定义, 491
- 连接启动的同步
  - 术语定义, 491
- 连接失败
  - UltraLite 故障排除, 42
- 连接条件
  - 术语定义, 491
- 连接字符串
  - UltraLite 参数概述, 42
  - UltraLite 字符串运算符, 317
  - UltraLite 设置, 43
  - UltraLite 连接参数, 207

## 链接

- UltraLite 引擎库, 19
- UltraLite 运行时库, 19

## 列

- UltraLite ALTER TABLE 语句, 428
- UltraLite 别名, 457
- UltraLite 变更方法, 66
- UltraLite 变更用法, 65
- UltraLite 复制方法, 69
- UltraLite 添加方法, 64
- UltraLite 限制, 6

## 列表

- UltraLite LIST 函数语法, 371

## 列名称

- UltraLite SQL 语法, 304

## 列压缩

- UltraLite SQL ALTER TABLE 语句, 428

## 临时表

- UltraLite 关于, 10
- UltraLite 同步使用, 124
- UltraLite 查询, 320
- UltraLite 管理, 108
- UltraLite 限制, 2, 6
- 术语定义, 491

## 临时文件

- UltraLite 限制, 10

## 零填充

- UltraLite date\_format 创建参数, 172

## 流参数

- UltraLite HotSync 同步, 133
- UltraLite 同步参数, 155

## 流错误

- UltraLite 同步参数, 152

## 流类型

- UltraLite 同步参数, 154

## 流同步参数

- UltraLite 同步参数, 155

## 路径

- UltraLite 引擎, 470
- UltraLite 连接参数, 40

## 轮询

- 术语定义, 492

## 逻辑

- UltraLite 为同步设计捕获, 123

## 逻辑索引

- 术语定义, 492

## 逻辑运算符

- UltraLite SQL 语法, 311

## 浏览

- UltraLite 表信息, 67
- UltraLite 表方法, 67

**M**

## max\_hash\_size 创建参数

- UltraLite 说明, 177

## max\_hash\_size 属性

- UltraLite 说明, 195

## MAX 函数

- UltraLite 语法, 376

## MINUTES 函数

- UltraLite 语法, 378

## MINUTE 函数

- UltraLite 语法, 377

## MIN 函数

- UltraLite 语法, 376

## MIRROR\_FILE 连接参数

- UltraLite 语法, 216

## ml\_add\_connection\_script 系统过程

- 添加, 96

## ml\_add\_table\_script 系统过程

- 添加, 96

## ML\_GET\_SERVER\_NOTIFICATION

- 语法, 379

## ml\_remote\_id

- UltraLite 属性配置, 245

- UltraLite 设置值, 205

## ml\_remote\_id 属性

- UltraLite 说明, 195

## ml\_remote\_id 选项

- UltraLite, 205

## mlasdesk.dll

- 部署 UltraLite 应用程序, 55

## mlasdev.dll

- 部署 UltraLite 应用程序, 55

## mlasinst 实用程序

- UltraLite 使用 DLL 文件进行部署, 55

## mlcecc11.dll

- 部署 UltraLite 应用程序, 132

## mlcrsa11.dll

- 部署 UltraLite 应用程序, 132

## mlcrsafips11.dll

- 部署 UltraLite 应用程序, 132

## mlczlib11.dll

- 部署 UltraLite 应用程序, 132

- MobiLink
    - UltraLite CREATE PUBLICATION 语句, 435
    - UltraLite SQL 语句, 422
    - UltraLite 客户端, 115
    - UltraLite 用户 ID 唯一性, 205
    - 术语定义, 492
  - MobiLink 服务器
    - UltraLite HotSync, 53
    - 术语定义, 492
  - MobiLink 监控器
    - 术语定义, 492
  - MobiLink 客户端
    - UltraLite 进度计数器, 116
    - 术语定义, 493
  - MobiLink 同步
    - UltraLite 客户端, 115
    - 设置 timestamp\_increment 创建参数, 192
  - MobiLink 文件传输
    - UltraLite 客户端概述, 129
  - MobiLink 系统表
    - 术语定义, 493
  - MobiLink 用户
    - 术语定义, 493
  - MOD 函数
    - UltraLite 语法, 380
  - MONEY 数据类型
    - UltraLite 等效项, 292
  - MONTHNAME 函数
    - UltraLite 语法, 381
  - MONTHS 函数
    - UltraLite 语法, 382
  - MONTH 函数
    - UltraLite 语法, 381
  - 密钥连接参数
    - UltraLite 语法, 214
  - 名称
    - UltraLite 列名称, 304
  - 名称属性
    - UltraLite 说明, 195
  - 命令 shell
    - 大括号, xv
    - 引号, xv
    - 括号, xv
    - 环境变量, xv
    - 约定, xv
  - 命令提示符
    - 大括号, xv
  - 引号, xv
  - 括号, xv
  - 环境变量, xv
  - 约定, xv
  - 命令文件
    - 术语定义, 492
  - 命令行实用程序
    - UltraLite HotSync 管道安装程序 [ulcond11] 实用程序, 243
    - UltraLite Interactive SQL [dbisql] 语法, 229
    - UltraLite Palm [ULDBUtil] 实用程序, 238
    - UltraLite SQL 预处理器 [sqlpp] 实用程序, 232
    - UltraLite 信息 [ulinfo] 实用程序, 245
    - UltraLite 初始化数据库 [ulinit] 实用程序, 249
    - UltraLite 卸载数据到 XML [ulunload] 实用程序, 260
    - UltraLite 卸载旧数据库 [ulunloadold] 实用程序, 263
    - UltraLite 同步 [ulsync], 255
    - UltraLite 引擎停止 [ulstop] 实用程序, 241
    - UltraLite 引擎启动 [uleng11] 实用程序, 240
    - UltraLite 数据库创建 [ulcreate] 实用程序, 235
    - UltraLite 消除数据库 [ulerase] 实用程序, 242
    - UltraLite 装载 XML 到数据库 [ulload] 实用程序, 252
  - 模糊处理
    - UltraLite 性能影响, 111
    - UltraLite 用法, 35
  - 模式
    - UltraLite SQL ALTER DATABASE SCHEMA FROM FILE 语法, 424
    - UltraLite 更改, 预防措施, 9
    - UltraLite 系统表, 267
    - UltraLite 表目录, 9
    - 术语定义, 493
  - 模式匹配
    - UltraLite PATINDEX 函数, 386
  - 目录
    - UltraLite 系统表, 267
- ## N
- NATURAL JOIN 子句
    - UltraLite 语法, 449
  - nearest\_century 创建参数
    - UltraLite 说明, 179
  - nearest\_century 属性
    - UltraLite 说明, 195

- new-database-ulcreatefile
  - UltraLite 数据库创建 [ulcreate] 实用程序, 235
- NEWID 函数
  - UltraLite 语法, 383
- newmobilinkpwd 同步参数
  - UltraLite 参考, 145
- nosync 表
  - UltraLite 概述, 63
  - UltraLite 非同步表, 124
- NOT
  - UltraLite 位运算符, 317
  - UltraLite 逻辑运算符, 311
- NOW 函数
  - UltraLite 语法, 384
- NT\_FILE 连接参数
  - UltraLite 语法, 218
- NULL
  - UltraLite ISNULL 函数, 367
  - 如果指定 NULL 参数则由函数返回, 323
- NULLIF 函数
  - UltraLite 关于, 384
  - 在 UltraLite 中与 CASE 表达式一起使用, 306
- NULL 值
  - UltraLite SQL, 285
- NUMERIC 数据类型
  - UltraLite, 290
- 内部引用
  - UltraLite SQL 的子查询, 307
- 内存
  - UltraLite 限制, 6
- 内存故障
  - UltraLite 防止, 224
- 内存使用
  - UltraLite 数据库存储, 40
  - UltraLite 索引, 100
  - UltraLite 行状态, 12
- 内连接
  - 术语定义, 493
- O**
- obfuscate 数据库创建参数
  - UltraLite 说明, 181
- obfuscate 数据库属性
  - UltraLite 用法, 35
- OBJECT\_ID 函数
  - 语法, 329
- OBJECT\_NAME 函数
  - 语法, 329
- observer 同步参数
  - UltraLite 说明, 146
- ODBC
  - 术语定义, 493
- ODBC 管理器
  - 术语定义, 493
- ODBC 数据源
  - 术语定义, 493
- OR
  - UltraLite 位运算符, 317
  - UltraLite 逻辑运算符, 311
- Oracle 统一数据库
  - 与其相关的 UltraLite 问题, 117
- ORDER BY 子句
  - UltraLite SELECT 语句, 457
  - UltraLite 故障排除, 476
- ORDERED\_TABLE\_SCAN 连接参数 (不建议使用)
  - UltraLite 语法, 219
- P**
- page\_size 创建参数
  - UltraLite 说明, 182
- page\_size 创建数据库属性
  - 早期版本的 Palm OS 的 UltraLite 优化, 112
- page\_size 属性
  - UltraLite 说明, 195
- PALM\_ALLOW\_BACKUP 连接参数
  - UltraLite 语法, 220
- PALM\_FILE 连接参数
  - UltraLite 语法, 221
- Palm Computing Platform
  - (参见 Palm OS)
- Palm HotSync 管道安装程序实用程序
  - 语法, 243
- Palm OS
  - UltraLite CustDB 示例和自述文件, 89
  - UltraLite HotSync, 53
  - UltraLite HotSync 部署, 53
  - UltraLite PDB 记录, 41
  - UltraLite VFS 数据库, 41
  - UltraLite VFS 部署, 236
  - UltraLite 发布限制, 124
  - UltraLite 字符集, 33
  - UltraLite 数据库, 41
  - UltraLite 数据管理实用程序, 238
  - 扩展卡, 使用, 41

- 用于 MobiLink 的 UltraLite 应用程序开发, 132
- 配置 UltraLite 针对早期版本, 112
- Palm 数据管理实用程序
  - 语法, 238
- partial download retained 同步参数
  - UltraLite 参考, 146
- password
  - UltraLite 同步参数, 147
- PATINDEX 函数
  - UltraLite 语法, 386
- PDB
  - (参见 UltraLite 数据库和 Palm OS)
  - UltraLite 数据库, 41
  - 术语定义, 493
- PDF
  - 文档, xii
- ping
  - UltraLite 同步参数, 148
- PI 函数
  - UltraLite 语法, 387
- PowerDesigner
  - 术语定义, 493
- PowerJ
  - 术语定义, 494
- POWER 函数
  - UltraLite 语法, 387
- publication
  - UltraLite 发布同步, 149
- PWD 连接参数
  - UltraLite 语法, 223
- 排序查询结果
  - UltraLite 关于, 108
- 排序顺序
  - UltraLite 归类, 33
- 派生表
  - UltraLite FROM 子句, 449
  - UltraLite SQL, 307
  - UltraLite SQL 的子查询, 307
- 批量装载
  - UltraLite LOAD TABLE 语句, 452
- 平方根函数
  - UltraLite SQRT 函数, 401
- 平均值函数
  - UltraLite AVG 函数, 335
- 平台
  - UltraLite 多个连接参数, 40
  - UltraLite 文件存储, 40

## Q

- QAnywhere
  - 术语定义, 494
- QAnywhere 代理
  - 术语定义, 494
- QUARTER 函数
  - UltraLite 语法, 388
- QUOTES 子句
  - UltraLite LOAD TABLE 语句, 453
- 前缀
  - 用于 Windows Mobile 的 UltraLite 数据库, 41
- 嵌入式 SQL
  - UltraLite 中的 NULL 值, 285
  - 术语定义, 494
- 请求
  - UltraLite 并发, 11
  - UltraLite 管理, 11
- 区分
  - UltraLite 大小写创建参数, 167
- 区分大小写
  - UltraLite 大小写创建参数, 167
  - UltraLite 大小写属性, 195
  - UltraLite 字符串, 282
  - UltraLite 的比较运算符, 310
- 区分大小写注意事项
  - UltraLite 关于, 167
- 全局临时表
  - 术语定义, 494
- 全局数据库 ID 注意事项
  - UltraLite 关于, 204
- 全局数据库标识符
  - UltraLite global\_database\_id, 204
  - UltraLite, 设置, 119
- 全局唯一标识符
  - MobiLink 系统中的 UltraLite 客户端, 119
  - NEWID 函数的 UltraLite SQL 语法, 383
- 全局自动增量
  - MobiLink 系统中的 UltraLite 客户端, 119
  - UltraLite global\_database\_id, 204
  - UltraLite, 设置, 119
  - UltraLite, 设置缺省值, 121
  - 在 UltraLite 中用完范围内的值, 119
- 缺省值
  - UltraLite CURRENT DATE, 286
  - UltraLite CURRENT TIME, 286
  - UltraLite CURRENT TIMESTAMP, 287
  - UltraLite SQLCODE, 287

UltraLite 自动增量, 440

## R

RADIANS 函数

UltraLite 语法, 389

RDBMS

术语定义, 486

ReadCommitted

UltraLite 隔离级别, 15

ReadUncommitted

UltraLite 隔离级别, 15

REAL 数据类型

UltraLite, 290

reload.sql

UltraLite 装载, 452

REMAINDER 函数

UltraLite 语法, 389

REMOTE DBA 权限

术语定义, 506

REPEAT 函数

UltraLite 语法, 390

REPLACE 函数

UltraLite 语法, 391

REPLICATE 函数

UltraLite 语法, 392

RESERVE\_SIZE 连接参数

UltraLite 语法, 224

resume partial download 同步参数

UltraLite 参考, 150

RIGHT OUTER JOIN 子句

UltraLite 语法, 449

RIGHT 函数

UltraLite 语法, 393

ROLLBACK 语句

UltraLite 语法, 456

ROUND 函数

UltraLite 语法, 394

rowlimit 操作

UltraLite 执行计划, 321

RSA 加密算法

UltraLite 启用 TLS 的同步, 51

RTRIM 函数

UltraLite 语法, 394

日期

UltraLite date\_format 属性, 195

UltraLite date\_order 属性, 195

UltraLite 不明确的字符串转换, 179

UltraLite 切换点, 179

UltraLite 排序, 174

UltraLite 格式, 171

UltraLite 转换函数, 324

日期部分

关于, 325

在 UltraLite 中可用, 171

日期函数

UltraLite 按字母顺序排序的列表, 324

日期时间

UltraLite 转换函数, 324

日期注意事项

UltraLite 关于, 171

日志文件

UltraLite Palm 同步, 134

术语定义, 494

## S

samples-dir

文档用法, xiv

scan 操作

UltraLite 执行计划, 321

scjview

安装前终止, 471

SECONDS 函数

UltraLite 语法, 396

SECOND 函数

UltraLite 语法, 395

SELECT 语句

UltraLite 复制结果集, 69

UltraLite 浏览数据示例, 68

UltraLite 结果集故障排除, 476

UltraLite 语法, 457

send\_column\_names 同步参数

UltraLite 参考, 151

send\_download\_ack 同步参数

UltraLite 参考, 152

setObserver 方法

UltraLite 示例, 146

SET OPTION 语句

UltraLite 语法, 459

setScriptVersion 方法

UltraLite 示例, 159

setStreamParms 方法

UltraLite 示例, 155

setStream 方法

UltraLite 示例, 154

setUserData 方法  
     UltraLite 示例, 158  
 SHORT\_PLAN 函数  
     UltraLite 语法, 397  
 数据库校验  
     UltraLite, 12  
 SIGN 函数  
     UltraLite 语法, 398  
 SIMILAR 函数  
     UltraLite 语法, 398  
 SIN 函数  
     UltraLite 语法, 399  
 SKIP 子句  
     UltraLite LOAD TABLE 语句, 453  
 SMALLINT 数据类型  
     UltraLite, 290  
 SMALLMONEY 数据类型  
     UltraLite 等效项, 292  
 SOUNDEX 函数  
     UltraLite 语法, 400  
 SPACE 函数  
     UltraLite 语法, 401  
 SQL  
     (参见 UltraLite SQL)  
     UltraLite 中的变量, 319  
     UltraLite 中的搜索条件, 309  
     UltraLite 中的数据类型, 290  
     UltraLite 中的表达式, 303  
     UltraLite 中的运算符, 316  
     UltraLite 保留字, 280  
     UltraLite 关键字, 280  
     UltraLite 字符串, 282  
     UltraLite 数字, 284  
     UltraLite 标识符, 281  
     UltraLite 模式更改, 9  
     UltraLite 的比较运算符, 310  
     UltraLite 语句类型, 423  
     术语定义, 498  
 SQL Anywhere  
     文档, xii  
     术语定义, 498  
 SQL Anywhere 数据库  
     与 UltraLite 进行数据库比较, 2  
 SQLCODE  
     UltraLite SQLE\_MAX\_ROW\_SIZE\_EXCEEDED  
     错误, 62  
     UltraLite 并发检查, 14  
     UltraLite 特殊值, 287  
 SQLCODE SQLE\_LOCKED  
     UltraLite 并发错误, 14  
 SQLE\_CONVERSION\_ERROR  
     UltraLite 升级警告, 59  
 SQLE\_DATABASE\_ERROR  
     UltraLite 数据损坏, 472  
 SQLE\_DEVICE\_ERROR  
     UltraLite 数据损坏, 472  
 SQLE\_DOWNLOAD\_CONFLICT 错误  
     UltraLite 同步, 127  
 SQLE\_MAX\_ROW\_SIZE\_EXCEEDED  
     UltraLite 错误, 62  
 SQLE\_MEMORY\_ERROR  
     UltraLite 数据损坏, 472  
 SQLE\_NOTFOUND  
     UltraLite 并发错误, 14  
 SQLE\_ROW\_DROPPED\_DURING\_SCHEMA\_UP  
 GRADE  
     UltraLite 升级警告, 58  
 SQLE\_UNABLE\_TO\_CONNECT\_OR\_START  
     故障排除, 470  
 SQL Flagger  
     UltraLite 用法, 232  
 sqlpp 实用程序  
     UltraLite 语法, 232  
 SQL Remote  
     术语定义, 498  
 SQL 代码  
     UltraLite 模式升级错误, 59  
 SQL 函数  
     UltraLite 函数按字母顺序排序的列表 A-D, 330  
     UltraLite 函数按字母顺序排序的列表 E-O, 359  
     UltraLite 函数按字母顺序排序的列表 P-Z, 386  
     UltraLite 函数的类型, 324  
     UltraLite 字符串, 327  
     UltraLite 数字, 327  
     UltraLite 数据类型转换, 324  
     UltraLite 日期和时间, 324  
     UltraLite 杂类, 326  
     UltraLite 简介, 323  
     UltraLite 系统, 328  
     UltraLite 集合, 324  
     如果指定参数 NULL 则返回 NULL, 323  
 SQL 语法  
     UltraLite IF 表达式, 305  
     UltraLite SQLCODE 特殊值, 287

- UltraLite 函数, 324
- UltraLite 列名称, 304
- UltraLite 常量, 304
- UltraLite 注释, 283
- UltraLite 特殊值, 286
- UltraLite 的 CASE 表达式, 305
- UltraLite 输入参数, 308
- SQL 语句
  - UltraLite, 422
  - 按字母顺序排序的 UltraLite 语句列表, 422
  - 术语定义, 498
- SQL 预处理器实用程序
  - UltraLite 语法, 232
- SQL 直通
  - UltraLite 支持, 83
- SQRT 函数
  - UltraLite 语法, 401
- START SYNCHRONIZATION DELETE 语句
  - UltraLite 语法, 460
- START 连接参数
  - UltraLite 语法, 225
  - Windows Mobile 上用于 uleng11 的 UltraLite, 49
- STOP SYNCHRONIZATION DELETE 语句
  - UltraLite 语法, 461
- stream\_error 同步参数
  - UltraLite ul\_stream\_error 结构, 152
  - UltraLite 参考, 152
- stream\_parms 同步参数
  - UltraLite 参考, 155
  - UltraLite 用法, 133
  - 使用 HotSync 的 UltraLite 同步, 132
- STRING 函数
  - UltraLite 语法, 403
- STRIP 子句
  - UltraLite LOAD TABLE 语句, 453
- STRTOUUID 函数
  - UltraLite 语法, 404
- STR 函数
  - UltraLite 语法, 402
- STUFF 函数
  - UltraLite 语法, 404
- subquery 操作
  - UltraLite 执行计划, 321
- SUBSCRIBE BY 子句
  - UltraLite 同步限制, 124
- SUBSTRING 函数
  - UltraLite 语法, 405
- SUBSTR 函数
  - UltraLite 语法, 405
- SUM 函数
  - UltraLite 语法, 406
- SUSER\_ID 函数
  - 语法, 407
- SUSER\_NAME 函数
  - 语法, 408
- Sybase Central
  - UltraLite 列创建方法, 64
  - UltraLite 创建发布, 125
  - UltraLite 创建索引, 72
  - UltraLite 创建表方法, 63
  - UltraLite 复制数据库对象方法, 69
  - UltraLite 系统表浏览方法, 67
  - UltraLite 表删除方法, 66
  - UltraLite 表变更方法, 65
  - UltraLite 表浏览方法, 67
  - UltraLite 连接故障排除, 471
  - 创建 UltraLite 数据库, 26
  - 术语定义, 499
  - 浏览 UltraLite 中的 CustDB, 96
- SYNC\_PROFILE\_OPTION\_VALUE 函数
  - 语法, 408
- SYNCHRONIZE 语句
  - UltraLite 语法, 462
- SYS
  - UltraLite 系统表, 267
  - 术语定义, 499
- sysarticle 系统表 [UltraLite]
  - 关于, 275
- syscolumn 系统表 [UltraLite]
  - 关于, 270
- sysindex 系统表 [UltraLite]
  - 关于, 271
- sysixcol 系统表 [UltraLite]
  - 关于, 273
- syspublication 系统表 [UltraLite]
  - 关于, 274
- sysstable 系统表 [UltraLite]
  - 关于, 269
- system\_error\_code 值
  - UltraLite 同步流错误, 153
- sysuldata 系统表 [UltraLite]
  - 关于, 276
- 散列
  - UltraLite max\_hash\_size 属性, 195



- UltraLite 大小注意事项, 177
- UltraLite 最佳大小, 105
- UltraLite 索引, 177
- UltraLite 配置大小, 107
- 术语定义, 494
- 扫描
  - 查询中的 UltraLite 数据库页, 108
  - 查询中的 UltraLite 索引, 100
- 删除
  - UltraLite DROP SYNCHRONIZATION PROFILE 语句, 447
  - UltraLite SQL CREATE INDEX 语句, 433
  - UltraLite SQL DROP INDEX 语句, 445
  - UltraLite SQL DROP PUBLICATION 语句, 446
  - UltraLite SQL DROP TABLE 语句, 448
  - UltraLite START SYNCHRONIZATION DELETE 语句, 460
  - UltraLite TRUNCATE TABLE 语句, 464
  - UltraLite 列, 428
  - UltraLite 发布, 77
  - UltraLite 数据库, 12
  - UltraLite 用户, 80
  - UltraLite 索引, 73
  - UltraLite 表方法, 66
  - 删除数据库的 UltraLite 实用程序, 238
- 删除 UltraLite 表
  - 关于, 66
- 删除 UltraLite 用户
  - 关于, 80
- 删除发布
  - UltraLite 客户端, 77
- 删除数据
  - UltraLite 文件大小影响, 473
- 删除索引
  - UltraLite 关于, 73
- 删除用户
  - UltraLite, 80
- 上载
  - 术语定义, 495
- 上载成功
  - UltraLite 同步参数, 156
- 舍入
  - UltraLite 小数位数, 185
  - UltraLite 小数位数创建参数, 185
- 设备
  - 设备跟踪
    - 术语定义, 495
  - 设备上创建
    - 关于, 29
  - 设计
    - UltraLite 实现, 17
  - 设计 UltraLite 中的同步
    - 关于, 123
  - 设置全局数据库标识符
    - MobiLink 系统中的 UltraLite 客户端, 119
  - 设置散列大小
    - 关于, 107
  - 声明缺省全局自动增量列
    - MobiLink 系统中的 UltraLite 客户端, 121
  - 生成的连接条件
    - 术语定义, 496
  - 升级
    - UltraLite SQL ALTER DATABASE SCHEMA FROM FILE 语法, 424
    - UltraLite 模式过程, 58
    - UltraLite 模式错误回调, 59
  - 升级 UltraLite
    - 连接故障排除, 471
  - 十六进制字符串
    - UltraLite 关于, 362
  - 时间
    - UltraLite 转换函数, 324
  - 时间戳
    - UltraLite timestamp\_format 创建参数, 189
    - UltraLite timestamp\_increment 创建参数, 192
  - 时间函数
    - UltraLite 按字母顺序排序的列表, 324
  - 时间注意事项
    - UltraLite 关于, 187
  - 实例化视图
    - 术语定义, 495
  - 实体关系图
    - UltraLite 关于, 69
  - 实体关系选项卡
    - UltraLite 使用, 69
  - 实用程序
    - ulcreate, 235
    - UltraLite HotSync 管道安装程序 [ulcond11] 实用程序, 243
    - UltraLite Interactive SQL [dbisql] 语法, 229
    - UltraLite Palm [ULDBUtil] 实用程序, 238
    - UltraLite SQL 预处理器 [sqlpp] 实用程序, 232

- UltraLite 信息 [ulinfo] 实用程序, 245
- UltraLite 初始化数据库 [ulinit] 实用程序, 249
- UltraLite 卸载数据到 XML [ulunload] 实用程序, 260
- UltraLite 卸载旧数据库 [ulunloadold] 实用程序, 263
- UltraLite 同步 [ulsync], 255
- UltraLite 引擎停止 [ulstop] 实用程序, 241
- UltraLite 引擎启动 [uleng11] 实用程序, 240
- UltraLite 故障排除, 475
- UltraLite 数据库创建 [ulcreate] 实用程序, 235
- UltraLite 消除数据库 [ulerase] 实用程序, 242
- UltraLite 装载 XML 到数据库 [ulload] 实用程序, 252
- UltraLite 错误代码, 228
- 设备上的 Windows Mobile 数据库管理, 41
- 使数据同步
  - UltraLite 文件大小影响, 473
- 使用索引
  - UltraLite 关于, 71
- 示例
  - (参见 教程)
  - (参见 示例)
- 示例应用程序
  - 在 UltraLite 中启动 CustDB, 91
- 世代号
  - 术语定义, 495
- 事件模型
  - 术语定义, 495
- 事件通知
  - UltraLite 使用, 81
- 事务
  - UltraLite COMMIT 语句, 432
  - UltraLite 刷新, 202
  - UltraLite 回退, 456
  - UltraLite 并发, 11
  - UltraLite 数据库, 12
  - UltraLite 检查点, 431
  - UltraLite 模式更改影响, 9
  - UltraLite 隔离级别, 15
  - 术语定义, 495
- 事务处理
  - UltraLite COMMIT\_FLUSH 连接参数, 210
  - UltraLite commit\_flush\_count 选项, 202
  - UltraLite commit\_flush\_timeout 选项, 203
  - 检查点和提交, 110
- 事务管理
  - UltraLite, 11
- 事务记录
  - UltraLite 内部机制, 13
- 事务日志
  - 术语定义, 495
- 事务日志镜像
  - 术语定义, 496
- 事务完整性
  - 术语定义, 496
- 视图
  - 术语定义, 495
- 授权选项
  - 术语定义, 496
- 受保护的功能
  - 术语定义, 496
- 受信任证书
  - UltraLite 应用程序访问加密信息, 256
- 输入参数
  - UltraLite 关于, 308
- 属性
  - DB\_PROPERTY 函数, 355
  - UltraLite 按字母顺序排序的列表, 195
  - UltraLite 数据库创建参数, 30
  - UltraLite 浏览, 199
  - UltraLite 系统表, 276
- 属性 (UltraLite)
  - DB\_PROPERTY 函数, 355
- 术语表
  - SQL Anywhere 术语列表, 481
- 数据
  - UltraLite 卸载, 260
  - UltraLite 查看方法, 67
  - UltraLite 装载, 252
  - UltraLite 选择行, 457
- 数据操作语言
  - 术语定义, 496
- 数据管理
  - UltraLite 状态信息, 11
  - UltraLite 组件可用于, 19
  - UltraLite 说明, 8
- 数据管理器
  - UltraLite 数据库存储, 40
- 数据库
  - (参见 UltraLite 数据库)
  - UltraLite 将批量数据装入, 452
  - 使用 UltraLite 插件创建, 26
  - 术语定义, 497

- 比较 UltraLite 和 SQL Anywhere, 2
- 数据库大小
  - UltraLite 限制, 6
- 数据库对象
  - UltraLite 复制方法, 69
  - 术语定义, 497
- 数据库服务器
  - 术语定义, 497
- 数据库管理
  - UltraLite 层, 8
- 数据库管理员
  - 术语定义, 497
- 数据库加密
  - UltraLite 性能影响, 111
- 数据库连接
  - 术语定义, 497
- 数据库名称
  - 术语定义, 497
- 数据库模式
  - UltraLite 系统表, 267
- 数据库实用程序
  - UltraLite 数据库连接, 43
- 数据库属性
  - (参见 数据库属性 (UltraLite))
  - UltraLite, 195
  - UltraLite 按字母顺序排序的列表, 195
- 数据库属性 (UltraLite)
  - fips 用法, 35
  - obfuscate 用法, 35
  - utf8\_encoding 用法, 33
  - 创建参数, 30
  - 浏览, 199
- 数据库所有者
  - 术语定义, 498
- 数据库文件
  - (参见 UltraLite 数据库)
  - UltraLite 加密, 214
  - UltraLite 最大大小, 6
  - UltraLite 连接参数, 40
  - 术语定义, 498
- 数据库选项
  - (参见 数据库选项 (UltraLite))
  - UltraLite, 201
- 数据库选项 (UltraLite)
  - commit\_flush\_count, 202
  - commit\_flush\_timeout, 203
  - DB\_PROPERTY 函数, 355
  - global\_database\_id, 204
  - ml\_remote\_id, 205
  - SET OPTION 语句, 459
  - 浏览, 206
- 数据库选项窗口
  - 访问, 206
- 数据库页面大小注意事项
  - UltraLite 关于, 182
- 数据库引擎
  - UltraLite 关于, 19
- 数据类型
  - UltraLite SQL 转换函数, 324
  - UltraLite 值转换, 292
  - UltraLite 别名等效项, 292
  - UltraLite 限制, 6
  - 按字母顺序排序的 UltraLite 数据类型列表, 290
  - 术语定义, 496
- 数据类型转换函数
  - UltraLite 关于, 324
- 数据立方体
  - 术语定义, 497
- 数据一致性
  - UltraLite 脏读, 15
- 数据源
  - UltraLite CustDB 教程示例, 94
- 数学表达式
  - UltraLite 算术运算符, 316
- 数字
  - UltraLite SQL, 284
- 数字函数
  - UltraLite 按字母顺序排序的列表, 327
- 数字精度
  - UltraLite 精度, 184
- 刷新
  - UltraLite 数据库, 202
- 刷新超时
  - UltraLite 格式, 203
- 刷新计数
  - UltraLite 格式, 202
- 双连字符
  - UltraLite 注释指示符, 283
- 双斜线
  - UltraLite 注释指示符, 283
- 死锁
  - UltraLite 防止, 14
  - 术语定义, 498
- 搜索条件

- UltraLite SQL, 309
- UltraLite SQL 的 ALL, 312
- UltraLite SQL 的 ANY, 313
- UltraLite SQL 的 BETWEEN, 313
- UltraLite SQL 的 EXISTS, 314
- UltraLite SQL 的 IN, 314
- 算术
  - 运算符和 UltraLite SQL 语法, 316
- 算术运算符
  - UltraLite SQL 语法, 316
- 索引
  - UltraLite page\_size 用法, 182
  - UltraLite sysindex 系统表, 271
  - UltraLite sysixcol 系统表, 273
  - UltraLite UNIQUE SQL 参数, 433
  - UltraLite 主键, 28
  - UltraLite 何时使用, 71
  - UltraLite 何时创建, 72
  - UltraLite 使用, 71
  - UltraLite 创建, 73
  - UltraLite 删除, 73
  - UltraLite 唯一索引特性, 72
  - UltraLite 唯一键特性, 72
  - UltraLite 复制方法, 69
  - UltraLite 性能增强, 103
  - UltraLite 散列值, 177
  - UltraLite 散列注意事项, 177
  - UltraLite 确定被扫描的索引, 102
  - UltraLite 简介, 100
  - UltraLite 类型, 72
  - UltraLite 跳过使用, 108
  - UltraLite 限制, 6
  - UltraLite 非唯一索引特性, 72
  - 术语定义, 499
- 索引扫描
  - UltraLite, 100
- 索引性能注意事项
  - UltraLite 关于, 177
- 锁定
  - UltraLite 并发, 14
  - 术语定义, 498
- T**
- tableOrder
  - UltraLite ulsync 选项, 255
- TableOrder
  - UltraLite 同步参数, 138
- TAN 函数
  - UltraLite 语法, 409
- TCP/IP
  - (参见 TCP/IP 同步)
- temp 操作
  - UltraLite 执行计划, 321
- TEXT 数据类型
  - UltraLite 等效项, 292
- THEN
  - UltraLite IF 表达式, 305
- 调整性能
  - 使用索引散列的 UltraLite, 103
- time\_format 创建参数
  - UltraLite 说明, 187
- time\_format 属性
  - UltraLite 说明, 195
- TIMESTAMP
  - UltraLite TIMESTAMP 列, 440
  - UltraLite 列限制, 2
- timestamp\_format 创建参数
  - UltraLite 说明, 189
- timestamp\_format 属性
  - UltraLite 说明, 195
- timestamp\_increment 创建参数
  - UltraLite 说明, 192
  - 在 MobiLink 同步中使用, 192
- timestamp\_increment 属性
  - UltraLite 说明, 195
- TIMESTAMP 数据类型
  - UltraLite, 290
- TIME 数据类型
  - UltraLite, 290
- TINYINT 数据类型
  - UltraLite, 290
- TLS
  - UltraLite 同步部署注意事项, 51
  - UltraLite 客户端配置, 51
- TLS\_TYPE 网络协议选项
  - UltraLite 启用 TLS 的同步, 51
- TODAY 函数
  - UltraLite 语法, 410
- TOP 子句
  - UltraLite SELECT 语句, 457
- TRIM 函数
  - UltraLite 语法, 410
- TRUNCATE TABLE 语句
  - UltraLite 语法, 464

---

TRUNCNUM 函数  
  UltraLite 语法, 411

TSEQUAL 函数  
  语法, 412

特殊值  
  UltraLite CURRENT DATE, 286  
  UltraLite CURRENT TIME, 286  
  UltraLite CURRENT TIMESTAMP, 287  
  UltraLite SQL, 286  
  UltraLite SQLCODE, 287

提供 UltraLite 连接参数  
  关于, 39

提供程序  
  UltraLite ActiveSync 文件, 55

提供程序文件  
  UltraLite 部署 ActiveSync, 55

提交  
  UltraLite COMMIT 语法, 432  
  UltraLite 事务概述, 14  
  UltraLite 数据库行, 12

提交刷新  
  UltraLite 配置, 202

提示  
  UltraLite 实现, 17

添加  
  UltraLite 列, 428  
  UltraLite 列方法, 64  
  UltraLite 用户, 79  
  UltraLite 索引, 73

添加 UltraLite 索引  
  关于, 72

添加 UltraLite 用户  
  关于, 79

添加同步  
  UltraLite 应用程序, 126

添加用户  
  UltraLite, 79

填充  
  使用 ulinit 创建的 UltraLite 数据库, 250

条件  
  UltraLite SQL 的 ALL 条件, 312  
  UltraLite SQL 的 ANY, 313  
  UltraLite SQL 的 BETWEEN, 313  
  UltraLite SQL 的 EXISTS, 314  
  UltraLite SQL 的 IN, 314  
  在 UltraLite SQL 中搜索, 309

通告程序  
  术语定义, 499

通配符  
  UltraLite PATINDEX 函数, 386

通信流  
  术语定义, 499

通用唯一标识符  
  (参见 UUID)  
  NEWID 函数的 UltraLite SQL 语法, 383

同步  
  UltraLite CustDB 教程, 94  
  UltraLite download\_only 参数, 142  
  UltraLite HotSync 协议选项, 133  
  UltraLite M-Business Anywhere 通道, 129  
  UltraLite Palm OS, 53  
  UltraLite SQLE\_DOWNLOAD\_CONFLICT 错误, 127  
  UltraLite 中的字符集, 32  
  UltraLite 仅上载参数, 157  
  UltraLite 任务概述, 117  
  UltraLite 使用发布排除表, 124  
  UltraLite 停止, 146  
  UltraLite 参照完整性, 126  
  UltraLite 只读表, 127  
  UltraLite 外键, 126  
  UltraLite 客户端, 115  
  UltraLite 客户端特定的数据, 124  
  UltraLite 应用程序实现, 126  
  UltraLite 忽略的行, 143  
  UltraLite 模式更改, 9  
  UltraLite 监控, 146  
  UltraLite 简介, 115  
  UltraLite 系统表, 269  
  UltraLite 设计概述, 123  
  UltraLite 进度计数机制, 116  
  Windows Mobile 上的 UltraLite 数据库, 135  
  术语定义, 499  
  用于 UltraLite 数据库的 ulsync 实用程序, 255  
  设置 timestamp\_increment 创建参数, 192

同步参数  
  UltraLite, 138  
  UltraLite additionalparms, 138  
  UltraLite Authentication Value, 142  
  UltraLite Disable Concurrency 概述, 11  
  UltraLite download\_only, 142  
  UltraLite getScriptVersion , 159  
  UltraLite getUploadOK 方法, 156  
  UltraLite keep partial download, 144

- UltraLite newmobilinkpwd, 145
- UltraLite observer , 146
- UltraLite partial download retained, 146
- UltraLite password, 147
- UltraLite ping, 148
- UltraLite publication, 149
- UltraLite resume partial download, 150
- UltraLite send\_column\_names, 151
- UltraLite send\_download\_ack , 152
- UltraLite setObserver 方法, 146
- UltraLite setScriptVersion 方法, 159
- UltraLite setSynchPublication 方法, 149
- UltraLite setUserData 方法, 158
- UltraLite stream\_error, 152
- UltraLite stream\_parms, 155
- UltraLite upload\_ok, 156
- UltraLite upload\_only, 157
- UltraLite user\_data, 158
- UltraLite user\_name, 158
- UltraLite version , 159
- UltraLite 中的 getStream 方法, 154
- UltraLite 中的 setStream 方法, 154
- UltraLite 中的 setStreamParms 方法, 155
- UltraLite 同步结果, 156
- UltraLite 必需的, 138
- UltraLite 流类型, 154
- 同步脚本
  - 浏览 UltraLite 示例, 96
- 同步结果
  - UltraLite 同步参数, 156
- 同步流
  - UltraLite setStream 方法, 154
  - UltraLite stream\_error 同步参数, 152
  - UltraLite ULHTTPStream, 154
  - UltraLite ULHTTPStream, 154
  - UltraLite 中的 getStream 方法, 154
  - UltraLite 中的 setStreamParms 方法, 155
  - UltraLite 中的 stream\_parms 同步参数, 155
  - UltraLite 流同步参数, 154
  - UltraLite 设置, 154
- 同步流参数
  - UltraLite 流类型, 154
- 同步逻辑
  - 浏览 UltraLite 中的 Sybase Central, 96
- 同步模型
  - UltraLite 数据库, 27
- 同步配置文件

- UltraLite ALTER SYNCHRONIZATION PROFILE 语句, 426
- UltraLite DROP SYNCHRONIZATION PROFILE 语句, 436, 447
- UltraLite SYNCHRONIZE 语句, 462
- 同步配置文件选项
  - UltraLite 同步, 258
  - 关于, 258
- 统一数据库
  - UltraLite 兼容性, 117
  - UltraLite 示例, 96
  - UltraLite 选择, 116
  - 术语定义, 499
- 图标
  - 此帮助文档中使用的, xv
- 图形式计划
  - 不支持 UltraLite, 320
- 推式请求
  - 术语定义, 500
- 推式通知
  - 术语定义, 500
- 退出代码
  - UltraLite 卸载数据到 XML [ulunload] 实用程序, 228
  - UltraLite 数据库创建 [ulcreate] 实用程序, 228
  - UltraLite 数据库同步 [ulsync] 实用程序, 228
  - UltraLite 的 Interactive SQL [dbisql] 实用程序, 231
  - UltraLite 装载 XML 到数据库 [ulload] 实用程序, 228

## U

- UCASE 函数
  - UltraLite 语法, 413
- UDB
  - (参见 UltraLite 数据库)
- UID 连接参数
  - UltraLite 语法, 226
- UL\_DEBUG\_CONDUIT\_LOG 环境变量
  - UltraLite HotSync 疑难解答, 134
- ul\_stream\_error 结构
  - UltraLite 示例, 152
- UL\_SYNC\_ALL\_PUBS 宏
  - UltraLite 发布列表, 149
- UL\_SYNC\_ALL 宏
  - UltraLite 发布列表, 149
- ulcond11 实用程序

- 
- 语法, 243
  - ULConduitStream 函数
    - UltraLite 同步流, 154
  - ulcreate 实用程序
    - UltraLite Palm OS VFS 注意事项, 236
    - 使用, 26
    - 语法, 235
  - ULDBUtil
    - 关于, 238
  - uleng11 实用程序
    - 语法, 240
    - 进程中数据库支持, 240
    - 部署到 Windows Mobile, 49
  - ulerase 实用程序
    - 语法, 242
  - ULHTTPStream 函数 [UL ESQL]
    - UltraLite 同步流, 154
  - ULHTTPStream 函数 [UL ESQL]
    - UltraLite 同步流, 154
  - ulinfo 实用程序
    - UltraLite 语法, 245
  - ulinit 实用程序
    - UltraLite Palm OS VFS 注意事项, 250
    - 不受支持的归类解决方案, 250
    - 使用, 27
    - 语法, 249
  - ulload 实用程序
    - UltraLite Palm OS VFS 注意事项, 254
    - 使用, 28
    - 语法, 252
  - ULSocketStream 函数
    - UltraLite 同步流, 154
  - ULSQLCONNECT 环境变量
    - 关于, 39
    - 说明, 46
  - ulstop 实用程序
    - 语法, 241
  - ulsync 实用程序
    - 同步配置文件选项, 258
    - 语法, 255
  - UltraLite
    - (参见 UltraLite API)
    - (参见 UltraLite SQL)
    - (参见 UltraLite 实用程序)
    - (参见 UltraLite 嵌入式 SQL)
    - (参见 UltraLite 应用程序)
    - (参见 UltraLite 数据库)
  - SQL 函数, 数据类型转换, 324
  - SQL 函数, 类型, 324
  - SQL 函数, 集合, 324
  - SQL 语句参考, 422
  - SQL\_MAX\_ROW\_SIZE\_EXCEEDED 错误, 62
  - 不受支持的归类, 250
  - 关于, 1
  - 多线程应用程序, 19
  - 安装应用程序和数据库, 47
  - 实用程序参考, 227
  - 故障排除, 469
  - 数据库创建参数, 165
  - 数据转换, 324
  - 术语定义, 500
  - 系统函数, 328
  - 表所有者, 281
  - 部署应用程序和数据库, 47
  - 错误代码, 228
  - UltraLite.NET
    - CustDB 示例和自述文件, 89
    - UltraLite 引擎支持, 19
    - 构建 CustDB 应用程序, 89
  - UltraLite API (参见 UltraLite C/C++ API) (参见 UltraLite for AppForge API) (参见 UltraLite for M-Business Anywhere API) (参见 UltraLite.NET API)
  - UltraLite C/C++
    - CustDB 示例和自述文件, 89
    - 引擎支持, 19
    - 构建 CustDB 应用程序, 89
  - UltraLite for M-Business Anywhere
    - UltraLite CustDB 示例和自述文件, 90
    - 引擎支持, 19
    - 构建 CustDB 应用程序, 90
    - 部署目标, 20
  - UltraLite HotSync 管道
    - 部署, 53
  - UltraLite 校验数据库实用程序 (ulvalid)
    - 关于, 265
  - UltraLite SQL
    - (参见 UltraLite 嵌入式 SQL)
    - CustDB 示例和自述文件, 89
    - NULL 值, 285
    - SQL 函数, 字符串, 327
    - SQL 函数, 数字, 327
    - SQL 函数, 杂类, 326
    - SQL 函数日期和时间, 324
    - UltraLite 按主键排序查询结果, 108
-

- 关键字, 280
- 变量, 319
- 基于索引的优化, 100
- 基于页的优化, 108
- 字符串, 282
- 执行计划, 320
- 数字, 284
- 数据类型, 290
- 日期, 289
- 时间, 289
- 查询故障排除, 476
- 标识符, 281
- 注释, 283
- 特殊值, 286
- 表达式, 303
- 运算符, 316
- 逗号分隔的列表, 371
- UltraLite SQL 语句
  - ALTER DATABASE SCHEMA FROM FILE 语句语法, 424
  - ALTER PUBLICATION 语句语法, 425
  - ALTER SYNCHRONIZATION PROFILE 语句语法, 426
  - ALTER TABLE 语句语法, 428
  - CHECKPOINT 语句语法, 431
  - COMMIT 语句语法, 432
  - CREATE INDEX 语句语法, 433
  - CREATE PUBLICATION 语句语法, 435
  - CREATE SYNCHRONIZATION PROFILE 语句语法, 436
  - CREATE TABLE 语句语法, 440
  - DELETE 语句语法, 444
  - DROP INDEX 语句语法, 445
  - FROM 子句, 449
  - INSERT 语句, 451
  - LOAD TABLE 语句语法, 452
  - ROLLBACK 语句语法, 456
  - SELECT 语句语法, 457
  - SET OPTION 语句语法, 459
  - START SYNCHRONIZATION DELETE 语句语法, 460
  - STOP SYNCHRONIZATION DELETE 语句语法, 461
  - SYNCHRONIZE 语句语法, 462
  - TRUNCATE TABLE 语法, 464
  - UltraLite DROP TABLE 语句语法, 448
  - UNION 操作语法, 466
  - UNIQUE 参数, 433
  - UPDATE 语句语法, 467
  - 关于, 422
  - 类别, 423
- UltraLite 表
  - 在模式升级过程中重命名, 58
  - 从创建过程中排除, 249
- UltraLite 插件
  - 故障排除, 471
- UltraLite 初始化数据库实用程序
  - 关于, 249
- UltraLite 创建参数
  - 关于, 165
- UltraLite 发布
  - 在模式升级过程中重命名, 58
- UltraLite 管理工具
  - 故障排除, 475
- UltraLite 客户端
  - 关于 MobiLink, 115
- UltraLite 客户端同步
  - 参数和选项关于, 138
- UltraLite 口令
  - 关于, 44
- UltraLite 连接
  - 故障排除, 470, 471
- UltraLite 连接参数
  - 关于, 42
  - 列表, 38
- UltraLite 列
  - 在模式升级过程中重命名, 58
- UltraLite 临时表
  - 管理, 108
- UltraLite 临时文件
  - 关于, 10
- UltraLite 模式
  - 设备上的升级, 58
- UltraLite 嵌入式 SQL
  - (参见 UltraLite 嵌入式 SQL 库函数)
  - CustDB 示例和自述文件, 89
  - 字符串, 232
  - 支持, 20
  - 行号, 232
  - 预处理器, 232
- UltraLite 实现
  - 关于, 17
- UltraLite 实用程序
  - dbisql 实用程序, 229



---

校验数据库实用程序, 265  
sqlpp 实用程序, 232  
ulcond11 实用程序, 243  
ulcreate 实用程序, 235  
ULDBUtil 实用程序, 238  
uleng10 实用程序, 240  
ulerase 实用程序, 242  
ulinfo 实用程序, 245  
ulinit 实用程序, 249  
ulload 实用程序, 252  
ulstop 实用程序, 241  
ulsync 实用程序, 255  
ulunload 实用程序, 260  
ulunloadold 实用程序, 263  
UltraLite 实用程序参考  
  关于, 227  
UltraLite 数据库  
  校验, 12  
  page\_size 创建参数, 182  
  page\_size 用法, 182  
  Palm OS 上的文件存储, 41  
  Palm OS 删除, 238  
  Palm OS 支持, 41  
  ULSQLCONNECT, 43  
  UltraLite 名称属性, 195  
  UltraLite 大小限制, 6  
  UltraLite 文件属性, 195  
  UltraLite 模式升级参考, 424  
  UNICODE 字符的 UTF8BIN 归类, 33  
  Windows Mobile 文件路径, 41  
  Windows 桌面操作系统, 41  
  在模式升级过程中重命名对象, 58  
  与 SQL Anywhere 进行数据库比较, 2  
  临时文件, 10  
  临时表和文件, 10  
  临时表管理, 108  
  事务概述, 11  
  从 MobiLink 建模, 27  
  从 Palm OS 设备删除应用程序数据, 238  
  从 SQL Anywhere 参考数据库创建, 27  
  从 Sybase Central 初始化, 26  
  从命令提示符创建, 26  
  从命令提示符初始化, 27  
  从系统故障恢复, 13  
  作为其参考数据库的 Oracle, 117  
  使用, 61  
  使用 [创建数据库向导] 创建, 26  
  使用 fips 创建参数加密, 176  
  使用的状态字节, 12  
  关于, 9  
  内存使用, 12  
  列变更, 65  
  列添加, 64  
  创建, 25  
  创建参数, 30  
  创建方法, 25  
  加密影响性能, 111  
  升级先前版本, 25  
  占用的资源, 2  
  发布关于, 75  
  发布删除, 77  
  发布行, 76  
  发布表, 75  
  同步实用程序 [ulsync] 语法, 255  
  同步简介, 11  
  同步计数, 116  
  同步配置文件选项, 258  
  唯一键, 72  
  回退, 12  
  在 Windows Mobile 上备份, 41  
  备份, 13  
  多表连接, 2  
  大小减小, 473  
  存储发布, 274  
  存储属性, 276  
  存储索引, 271  
  安全概述, 35  
  实体关系图, 69  
  实现, 17  
  实用程序参考, 227  
  对象复制方法, 69  
  属性优化策略, 112  
  属性浏览, 199  
  嵌入式引擎客户端文件, 19  
  并发, 11  
  建立主键索引, 28  
  引擎停止, 241  
  引擎启动, 240  
  引擎和运行时, 19  
  归类序列, 32  
  恢复, 12  
  描述发布, 275  
  描述表, 269  
  描述表的列, 273

- 支持的索引类型, 100
- 支持的网络协议, 18
- 故障排除, 472, 473, 474
- 数据和状态管理, 11
- 数据库完整性, 12
- 数据管理选项, 19
- 文件内部, 9
- 文件名过滤器, 238
- 文件路径定义, 40
- 显示系统表, 268
- 查看选项设置, 206
- 桌面操作系统创建选项, 26
- 检查点用法, 110
- 模式, 267
- 模式更改, 9
- 模式概述, 9
- 模糊处理对性能影响, 111
- 死锁, 14
- 消除数据库, 242
- 源自 XML, 28
- 环境变量, 43
- 用户 ID, 44
- 用户删除, 80
- 用户添加, 79
- 简介, 8
- 管理基础, 11
- 管理多个, 11
- 管理层, 8
- 系统故障恢复, 13
- 索引, 何时创建, 72
- 索引使用, 71
- 索引创建, 433
- 索引散列, 177
- 索引散列概述, 177
- 索引类型, 72
- 线程概述, 11
- 组成, 9
- 维护行状态, 12
- 行删除, 12
- 行压缩, 182
- 行压缩简介, 62
- 行大小限制, 62
- 行读取, 15
- 行锁定, 14
- 表, 浏览, 67
- 表创建, 62
- 表删除, 66
- 表同步后缀, 63
- 表复制, 68
- 表定义提示, 62
- 表过滤, 68
- 设备上的升级模式, 58
- 请求概述, 11
- 读取行, 15
- 运行 ulinit 后填充, 250
- 运行时文件, 19
- 连接优化策略, 112
- 连接参数列表, 38
- 连接参数概述, 42
- 连接故障排除, 471
- 连接概述, 11, 37
- 部署到设备, 47
- 部署同步加密, 51
- 部署数据加密, 50
- 部署选项, 19
- 隔离级别, 15
- UltraLite 数据库创建实用程序
  - 语法, 235
- UltraLite 数据库属性
  - 关于, 195
- UltraLite 数据库同步实用程序
  - 同步配置文件选项, 258
- UltraLite 同步
  - 关于, 115
  - 远程 ID 和用户 ID, 205
- UltraLite 同步实用程序
  - 语法, 255
- UltraLite 系统表
  - 关于, 267
- UltraLite 系统表参考
  - 关于, 267
- UltraLite 消除数据库实用程序
  - 语法, 242
- UltraLite 卸载旧数据库实用程序
  - 语法, 263
- UltraLite 卸载数据库实用程序
  - 语法, 260
- UltraLite 信息实用程序
  - 关于, 245
- UltraLite 引擎
  - Windows Mobile 部署, 49
  - 与进程中运行时相比, 19
  - 停止实用程序, 241
  - 启动实用程序, 240

---

- 在连接时定义可执行文件的位置, 225
- 并发, 11
- 故障排除, 470
- 数据管理, 19
- 消除数据库实用程序为了, 242
- 错误 -764 故障排除, 477
- 错误代码, 228
- UltraLite 引擎启动实用程序
  - 语法, 240
- UltraLite 引擎停止实用程序
  - 语法, 241
- UltraLite 应用程序
  - (参见 UltraLite C/C++ API)
  - (参见 UltraLite for AppForge API)
  - (参见 UltraLite for M-Business Anywhere API)
  - (参见 UltraLite.NET API)
  - API 选择, 20
  - CustDB 应用程序和自述文件, 87
  - SQL\_E\_UNABLE\_TO\_CONNECT\_OR\_START 故障排除, 470
  - 使用 MobiLink 传输文件, 129
  - 公共证书访问, 256
  - 同步, 115
  - 启用 TLS 的同步, 51
  - 在连接时定义引擎的位置, 225
  - 并发, 11
  - 库选择, 20
  - 开发平台, 20
  - 支持的 Windows 平台, 20
  - 构建 CustDB, 89
  - 管理多个请求, 11
  - 部署 ActiveSync 提供程序文件, 55
  - 部署 HotSync 管道文件, 132
  - 部署到设备, 49
  - 部署启用了 FIPS, 176
  - 错误 -764, 477
- UltraLite 用户 ID
  - MobiLink 唯一性, 205
  - 关于, 44
- UltraLite 优化程序
  - UltraLite 执行计划访问选项, 102
- UltraLite 运行时
  - 关于, 19
  - 并发, 11
  - 术语定义, 500
- UltraLite 中的表达式
  - 关于, 303
- UltraLite 中的数据类型
  - 关于, 290
- UltraLite 装载 XML 到数据库实用程序
  - 语法, 252
- ulunloadold 实用程序
  - 语法, 263
- ulunload 实用程序
  - 语法, 260
- ulvalid
  - 关于, 265
- UNICODE 字符
  - UltraLite 归类, 33
- union-all 操作
  - UltraLite 执行计划, 321
- UNION 操作
  - UltraLite 语法, 466
- UNION 语句
  - UltraLite 语法, 466
- UNIQUE
  - UltraLite CREATE INDEX 参数, 433
- UNIQUEIDENTIFIER 数据类型
  - UltraLite, 290
- UPDATE 语句
  - UltraLite 语法, 467
- UpgradeSchemaFromFile 方法
  - UltraLite 替换模式升级, 58
- upload\_ok 同步参数
  - UltraLite 参考, 156
- upload\_only 同步参数
  - UltraLite 参考, 157
- UPPER 函数
  - UltraLite 语法, 414
- user\_data
  - UltraLite 同步参数, 158
- USER\_ID 函数
  - 语法, 414
- user\_name
  - UltraLite 同步参数, 158
- USER\_NAME 函数
  - 语法, 415
- utf8\_encoding 创建参数
  - UltraLite 说明, 193
- utf8\_encoding 数据库属性
  - UltraLite 用法, 33
- UTF8BIN 归类
  - UltraLite 注意事项, 33
- UUID

NEWID 函数的 UltraLite SQL 语法, 383  
STRTOUUID 函数的 UltraLite SQL 语法, 404  
UIDTOSTR 函数的 UltraLite SQL 语法, 416  
UIDTOSTR 函数  
UltraLite 语法, 416

## V

VARBINARY 数据类型  
UltraLite, 290  
VARCHAR 数据类型  
UltraLite, 290  
version 同步参数  
UltraLite 参考, 159  
VFS  
UltraLite 数据库, 41  
UltraLite 数据库部署, 236  
Visual Basic 兼容性  
UltraLite 支持, 20  
Visual Studio  
UltraLite 构建 CustDB 应用程序, 89

## W

WEEKS 函数  
UltraLite 语法, 417  
WHEN  
UltraLite 的 CASE 表达式, 305  
WHERE 子句  
UltraLite CREATE PUBLICATION 语句, 435  
UltraLite DELETE 语句, 444  
UltraLite SELECT 语句, 457  
UltraLite UPDATE 语句, 467  
UltraLite 发布用法, 76  
UltraLite 同步限制, 124  
Windows  
(参见 Windows ME)  
(参见 Windows Mobile 5)  
(参见 Windows NT)  
(参见 Windows Vista)  
(参见 Windows XP/200x)  
UltraLite 字符集, 33  
术语定义, 502  
Windows Mobile  
UltraLite ActiveSync 部署, 55  
UltraLite FIPS 启用, 176  
UltraLite MobiLink 客户端, 135  
UltraLite uleng11 部署, 49  
UltraLite 使用 .NET 构建 CustDB 应用程序, 89

UltraLite 字符集, 33  
UltraLite 引擎支持, 19  
UltraLite 引擎部署, 49  
UltraLite 文件路径前缀, 41  
术语定义, 502  
错误 -764 故障排除, 477  
Windows 桌面操作系统  
UltraLite 引擎支持, 19  
UltraLite 数据库, 41  
WITH CHECKPOINT 子句  
UltraLite LOAD TABLE 语句, 454  
外表  
术语定义, 500  
外部登录  
术语定义, 500  
外部引用  
UltraLite SQL 的子查询, 307  
外键  
UltraLite 复制方法, 69  
UltraLite 外键, 440  
UltraLite 特性, 100  
未命名的 UltraLite, 440  
术语定义, 500  
外键循环  
UltraLite 关于, 126  
UltraLite 问题, 126  
外键约束  
术语定义, 501  
外连接  
术语定义, 501  
完全备份  
术语定义, 501  
完整性  
UltraLite CREATE TABLE 语句, 440  
术语定义, 501  
完整性约束  
UltraLite 用法, 440  
网关  
术语定义, 501  
网络服务器  
术语定义, 501  
网络协议  
UltraLite 同步结果同步参数, 156  
UltraLite 支持的, 18  
使用 HTTP 的 UltraLite 同步, 154  
使用 HTTPS 的 UltraLite 同步, 154  
使用 TCP/IP 的 UltraLite 同步, 154

- 术语定义, 502
  - 唯一键
    - UltraLite 特性, 100
    - UltraLite 索引创建, 72
  - 唯一索引
    - UltraLite UNIQUE SQL 参数, 433
    - UltraLite 特性, 100
    - UltraLite 索引创建, 72
  - 唯一约束
    - UltraLite CREATE TABLE 语句, 440
    - UltraLite 复制方法, 69
    - UltraLite 特性, 100
    - 术语定义, 502
  - 维护
    - UltraLite 在设备上, 47
  - 维护版本
    - 术语定义, 502
  - 维护主键的唯一性
    - MobiLink 系统中的 UltraLite 客户端, 119
  - 未命名的外键
    - UltraLite 用法, 440
  - 未提交事务
    - UltraLite 概述, 14
    - UltraLite 隔离级别, 15
  - 位数
    - UltraLite 最大数, 184
  - 位数组
    - 术语定义, 502
  - 位运算符
    - UltraLite SQL 语法, 317
  - 谓词
    - UltraLite SQL, 309
    - UltraLite SQL 的 ALL, 312
    - UltraLite SQL 的 ANY, 313
    - UltraLite SQL 的 BETWEEN, 313
    - UltraLite SQL 的 EXISTS, 314
    - UltraLite SQL 的 IN, 314
    - UltraLite 的比较运算符, 310
    - 术语定义, 502
  - 文本计划
    - 在 UltraLite 中查看, 320
  - 文档
    - SQL Anywhere, xii
    - 约定, xiii
  - 文件
    - HotSync 管道, 132
    - UltraLite ActiveSync 提供程序, 55
    - UltraLite CustDB 示例位置, 87
    - 使用 MLFileTransfer 传输, 129
  - 文件大小
    - UltraLite 数据库故障排除, 473
  - 文件定义数据库
    - 术语定义, 502
  - 文件对象
    - UltraLite 类型, 9
  - 文件名
    - UltraLite 连接参数, 40
  - 文件属性
    - UltraLite 说明, 195
  - 文件系统
    - (参见 VFS)
  - 文字
    - UltraLite 常量, 304
  - 物理索引
    - 术语定义, 503
  - 物理限制
    - UltraLite, 6
- ## X
- XML
    - UltraLite 数据库源自, 28
    - 卸载数据库到, 255
    - 装载到数据库, 252
  - XML 数据类型
    - UltraLite 等效项, 292
  - 系统表
    - UltraLite sysarticle, 275
    - UltraLite syscolumn, 270
    - UltraLite sysindex, 271
    - UltraLite sysixcol, 273
    - UltraLite syspublication, 274
    - UltraLite systable, 269
    - UltraLite sysuldata, 276
    - UltraLite 关于, 267
    - UltraLite 浏览方法, 67
    - UltraLite 隐藏和显示, 268
    - 术语定义, 503
  - 系统对象
    - UltraLite 显示方法, 68
    - 术语定义, 503
  - 系统故障
    - UltraLite 事务概述, 14
    - UltraLite 恢复, 13
  - 系统函数

- UltraLite 关于, 328
- UltraLite 限制, 2
- 系统视图
  - 术语定义, 503
- 下载
  - 术语定义, 503
- 下载确认
  - UltraLite send\_download\_ack 同步参数, 152
- 限制
  - UltraLite, 6
  - UltraLite 数据类型, 290
- 线程
  - UltraLite 并发, 11
- 相关名
  - 术语定义, 503
- 项
  - ulcond11 高速缓存大小的注册表, 243
- 项目
  - UltraLite 复制方法, 69
  - UltraLite 数据库, 124
  - UltraLite 限制, 124
  - 术语定义, 503
- 消除
  - UltraLite 表方法, 66
- 消息存储库
  - 术语定义, 503
- 消息类型
  - 术语定义, 503
- 消息日志
  - 术语定义, 504
- 消息系统
  - 术语定义, 504
- 小数
  - 在 UltraLite 中使用, 284
- 小数点位置注意事项
  - UltraLite 小数位数创建参数, 185
  - UltraLite 精度, 184
- 小数精度
  - UltraLite 精度, 184
- 小数位数创建参数
  - UltraLite 说明, 185
- 小数位数属性
  - UltraLite 说明, 195
- 小写字符串
  - UltraLite LCASE 函数, 368
  - UltraLite LOWER 函数, 374
- 校验
  - UltraLite checksum\_level 创建参数, 168
- 校验和
  - UltraLite checksum\_level 创建参数, 168
  - UltraLite checksum\_level 属性, 195
- 协议选项
  - UltraLite HotSync, 133
- 斜线加星号
  - UltraLite 注释指示符, 283
- 卸载
  - UltraLite 数据库, 260
  - 使用 ulunload 卸载 UltraLite 数据库, 260
  - 使用 ulunloadold 卸载 UltraLite 数据库, 263
  - 早期版本的 UltraLite 数据库, 263
  - 术语定义, 504
- 卸载旧数据库实用程序 [ulunloadold]
  - 语法, 263
- 新口令
  - UltraLite 同步参数, 145
- 新闻组
  - 技术支持, xvii
- 星期
  - UltraLite DOW 函数, 357
- 行
  - UltraLite INSERT 语句, 451
  - UltraLite 从表中删除所有行, 464
  - UltraLite 发布, 76
  - UltraLite 批量插入, 452
  - UltraLite 读取, 15
  - UltraLite 锁定, 14
- 行长度
  - UltraLite sqlpp 实用程序输出, 232
- 行级触发器
  - 术语定义, 487
- 行压缩
  - UltraLite 关于, 62
  - UltraLite 的影响, 182
  - UltraLite 观察, 224
- 性能
  - UltraLite CACHE\_SIZE 连接参数, 208
  - UltraLite ulcond11 高速缓存, 243
  - UltraLite 仅上载同步, 157
  - UltraLite 仅下载同步, 142
  - UltraLite 使用索引, 71
  - UltraLite 使用索引提高查询性能, 433
  - UltraLite 对应用程序使用索引, 28
  - UltraLite 将索引用于大表, 71
  - UltraLite 指定 FOR READ ONLY 子句, 320

- UltraLite 查看执行计划, 320
- UltraLite 查询优化, 458
- UltraLite 查询优化技术, 99
- UltraLite 索引散列, 177
- UltraLite 选择最佳的散列大小, 105
- UltraLite 防止内存故障, 224
- UltraLite 页面大小, 182
- 从检查点分隔提交, 110
- 使用索引散列的 UltraLite 查询调整, 103
- 性能统计
  - 术语定义, 504
- 修改
  - UltraLite 列, 428
  - UltraLite 表, 428
- 虚拟文件系统 (见 VFS)
- 序列化事务
  - UltraLite 处理, 14
- 选项
  - UltraLite 浏览, 206
- 选项 (UltraLite)
  - commit\_flush\_count, 202
  - commit\_flush\_timeout, 203
  - DB\_PROPERTY 函数, 355
  - global\_database\_id, 204
  - ml\_remote\_id, 205
  - SET OPTION 语句, 459
- 选择
  - UltraLite SELECT 语句, 457
- 选择编程接口
  - UltraLite 关于, 20
- 选择索引类型
  - UltraLite 关于, 72
- 选择一个数据管理组件
  - UltraLite 关于, 19
- 循环
  - UltraLite 外键, 126
  - UltraLite 外键问题, 126
- Y**
- YEARS 函数
  - UltraLite 语法, 419
- YEAR 函数
  - UltraLite 语法, 418
- YMD 函数
  - UltraLite 语法, 420
- 压缩的列
  - UltraLite ALTER TABLE 语句, 428

- 压缩行
  - UltraLite 关于, 62
  - UltraLite 的影响, 182
- 延迟提交
  - 性能增强, 110
- 演示
  - (参见 教程)
- 验证
  - UltraLite 设置, 44
  - UltraLite 跳过, 44
- 验证参数
  - UltraLite 同步参数, 140
- 验证参数的数目
  - UltraLite 同步参数, 145
- 验证值
  - UltraLite 同步参数, 142
- 验证状态
  - UltraLite 同步参数, 141
- 页面
  - UltraLite 大小注意事项, 182
- 页面大小
  - UltraLite page\_size 属性, 195
- 业务规则
  - 术语定义, 504
- 疑难解答
  - UltraLite getUploadOK 方法, 156
  - UltraLite HotSync, 134
  - UltraLite ping 同步参数, 148
  - UltraLite Stream Error 同步参数, 152
  - UltraLite Sync Result 同步参数, 156
  - UltraLite upload\_ok 同步参数, 156
  - UltraLite 全局 ID 号, 120
  - UltraLite 备份应用程序, 115
  - UltraLite 实现断点续传下载, 128
  - UltraLite 更改归类, 32
  - UltraLite 检索 GLOBAL AUTOINCREMENT 值, 120
  - UltraLite 连接参数优先级, 43
  - 在早期版本的 Palm OS 上运行 UltraLite, 112
  - 新闻组, xvii
  - 避免与外键循环相关的同步问题, 126
- 以 10 为底的对数
  - UltraLite LOG10 函数, 373
- 异或
  - UltraLite 位运算符, 317
- 引擎
  - (参见 UltraLite 引擎)

- 引用对象
  - 术语定义, 504
- 应用程序
  - (参见 UltraLite 应用程序)
- 用户
  - UltraLite 使用, 79
  - UltraLite 删除, 80
  - UltraLite 添加, 79
- 用户 ID
  - UltraLite 数据库, 44
  - UltraLite 更改, 79
  - UltraLite 注意事项, 79
  - UltraLite 添加新的, 44
  - UltraLite 缺省值, 79
  - UltraLite 语义, 44
- 用户定义的数据类型
  - UltraLite 中不支持的, 290
  - UltraLite 等效项, 292
- 用户定义数据类型
  - 术语定义, 505
- 用户名
  - UltraLite 同步参数, 158
- 用户数据
  - UltraLite 同步参数, 158
- 用户验证
  - PWD UltraLite 连接参数, 223
  - UltraLite Authentication Value 同步参数, 142
  - UltraLite getUsername 方法, 158
  - UltraLite password 同步参数, 147
  - UltraLite user\_name 同步, 158
  - UltraLite 作用, 44
  - UltraLite 关于, 44
  - UltraLite 同步状态报告, 141
  - UltraLite 同步自定义, 145
  - UltraLite 设置, 44
  - UltraLite 跳过, 44
  - 自定义 MobiLink, 140
- 用户验证的作用
  - UltraLite 关于, 44
- 优化
  - UltraLite SQL, 320
  - UltraLite 执行计划访问选项, 102
  - UltraLite 查询, 458
  - UltraLite 检查点, 110
  - UltraLite 索引, 105
- 优化 UltraLite
  - 查询性能, 99
- 优化程序
  - (参见 查询优化程序)
  - UltraLite 使用, 320
  - UltraLite 影响, 320
  - UltraLite 执行计划访问选项, 102
  - UltraLite 替换, 320
  - UltraLite 计划操作, 321
  - UltraLite 计划解释, 321
- 优先级
  - UltraLite SQL 运算符优先级, 317
- 游标
  - UltraLite 当前行, 11
  - UltraLite 脏读, 15
  - 术语定义, 505
- 游标结果集
  - 术语定义, 505
- 游标位置
  - 术语定义, 505
- 余切函数
  - UltraLite COT 函数, 345
- 余弦函数
  - UltraLite COS 函数, 344
- 语法
  - UltraLite CURRENT DATE 特殊值, 286
  - UltraLite CURRENT TIMESTAMP 特殊值, 287
  - UltraLite IF 表达式, 305
  - UltraLite SQL CURRENT TIME 特殊值, 286
  - UltraLite SQL 函数, 324
  - UltraLite SQL 函数 A-D, 330
  - UltraLite SQL 函数 E-O, 359
  - UltraLite SQL 函数 P-Z, 386
  - UltraLite SQL 注释, 283
  - UltraLite SQL 输入参数, 308
  - UltraLite SQL 运算符, 316
  - UltraLite SQL 运算符优先级, 317
  - UltraLite SQLCODE 特殊值, 287
  - UltraLite 位运算符, 317
  - UltraLite 列名称, 304
  - UltraLite 字符串运算符, 317
  - UltraLite 常量, 304
  - UltraLite 特殊值, 286
  - UltraLite 的 CASE 表达式, 305
  - UltraLite 的比较运算符, 310
  - UltraLite 算术运算符, 316
  - UltraLite 逻辑运算符, 311
- 语句
  - UltraLite, 422



- UltraLite 类型, 423
- UltraLite 预准备, 输入参数, 308
- 语句级触发器
  - 术语定义, 505
- 域
  - 术语定义, 505
- 预订
  - 术语定义, 506
- 预准备语句
  - UltraLite 输入参数, 308
- 元数据
  - UltraLite 考虑保留大小, 224
  - 术语定义, 506
- 原子事务
  - 术语定义, 506
- 远程 ID
  - 在 UltraLite 数据库中设置, 205
  - 术语定义, 506
- 远程服务器
  - UltraLite CREATE TABLE 语句, 440
- 远程数据库
  - UltraLite 同步计数, 116
  - 创建 UltraLite 客户端, 118
  - 删除 UltraLite 数据, 238
  - 术语定义, 506
- 约定
  - 命令 shell, xv
  - 命令提示符, xv
  - 文档, xiii
  - 文档中的文件名, xiv
- 约束
  - UltraLite ALTER TABLE 语句, 428
  - UltraLite 参照完整性, 126
  - UltraLite 重命名, 428
  - 术语定义, 507
- 运算符
  - UltraLite SQL 语法, 316
  - UltraLite 位运算符, 317
  - UltraLite 字符串运算符, 317
  - UltraLite 的比较运算符, 310
  - UltraLite 算术运算符, 316
  - UltraLite 运算符优先级, 317
  - UltraLite 逻辑运算符, 311
- 运算符优先级
  - UltraLite SQL 语法, 317
- 运算顺序
  - UltraLite SQL 运算符优先级, 317

- 运行
  - UltraLite CustDB 应用程序, 89
- 运行时
  - (参见 UltraLite 运行时)
- 运行时库
  - UltraLite 列表, 19
- 运营公司
  - 术语定义, 507

## Z

- zero-padding
  - UltraLite timestamp\_format 创建参数, 190
- 脏读
  - UltraLite 隔离级别, 15
- 增量备份
  - 术语定义, 507
- 占位符
  - UltraLite SQL 输入参数, 308
- 占用的资源
  - UltraLite 数据库, 2
- 争用
  - 术语定义, 507
- 整个表
  - UltraLite 发布, 125
- 正则表达式
  - 术语定义, 507
- 证书
  - UltraLite 应用程序访问加密信息, 256
- 支持
  - 新闻组, xvii
- 直方图
  - 术语定义, 507
- 直接行处理
  - 术语定义, 507
- 直接页扫描
  - UltraLite 关于, 108
- 直通模式
  - UltraLite, 83
- 执行计划
  - UltraLite 使用, 320
  - UltraLite 如何读取, 321
  - UltraLite 操作, 321
  - UltraLite 文本, 320
  - UltraLite 替换, 320
  - UltraLite 检查索引用法, 102
- 值
  - UltraLite 索引散列, 177

- 指示符
  - SQL 块中的 UltraLite 注释, 283
- 指数
  - 在 UltraLite 中, 284
- 指数函数
  - UltraLite EXP 函数, 359
- 只读表
  - UltraLite 同步, 127
  - UltraLite 数据库, 127
- 终止进程
  - UltraLite 升级故障排除, 471
- 主表
  - 术语定义, 508
- 主机名
  - UltraLite ULSynchronize 参数, 155
- 主机平台
  - UltraLite Windows 支持的平台, 20
- 主键
  - UltraLite UUID 和 GUID, 383
  - UltraLite 使用 UUID 生成唯一值, 383
  - UltraLite 列顺序, 440
  - UltraLite 完整性约束, 440
  - UltraLite 数据导入故障排除, 474
  - UltraLite 特性, 100
  - UltraLite 生成唯一值, 383
  - UltraLite 索引建立, 28
  - UltraLite 表, 63
  - UltraLite 表顺序, 126
  - 术语定义, 508
- 主键索引
  - UltraLite 使用, 108
  - UltraLite 跳过使用, 108
- 主键约束
  - 术语定义, 508
- 主题
  - 图标, xv
- 注册
  - UltraLite Hotsync 键, 53
  - 使用 ActiveSync 注册 MobiLink UltraLite 应用程序, 57
- 注册表项
  - ulcond11 高速缓存大小, 243
- 注释
  - UltraLite 语法, 283
- 转换
  - UltraLite 不明确的日期, 179
  - UltraLite 数据类型列表, 292
- 转换函数
  - UltraLite 按字母顺序排序的列表, 324
- 转换数据类型
  - UltraLite 关于, 292
- 转换字符串
  - UltraLite 关于, 327
- 转义序列
  - UltraLite 引擎路径, 477
- 装载
  - UltraLite LOAD TABLE 语句, 452
  - UltraLite 批量插入, 452
- 装载数据
  - UltraLite LOAD TABLE 语句, 452
- 装载数据库
  - 使用 unload 装载 UltraLite 数据库, 252
- 状态管理
  - UltraLite 概述, 11
- 状态字节
  - UltraLite 数据库, 12
- 桌面操作系统创建
  - UltraLite 关于, 26
- 子查询
  - UltraLite SQL, 307
  - 术语定义, 508
- 子串
  - UltraLite 关于, 405
- 子字符串
  - UltraLite 替换, 391
- 自动提交
  - UltraLite 事务概述, 14
- 自然连接
  - 术语定义, 496
- 自述文件
  - UltraLite CustDB 应用程序, 87
- 字
  - UltraLite 保留字, 280
  - UltraLite 关键字, 280
- 字符串
  - UltraLite nearest\_century 到日期的转换使用, 179
  - UltraLite SQL, 282
  - UltraLite SQL 函数, 327
  - UltraLite 删除尾随空白, 394
  - UltraLite 区分大小写, 282
  - UltraLite 嵌入式 SQL, 232
  - UltraLite 替换, 391
  - UltraLite 最大大小, 6
  - 术语定义, 508

---

- 字符串长度
  - UltraLite LENGTH 函数, 370
- 字符串函数
  - UltraLite 按字母顺序排序的列表, 327
- 字符串位置
  - UltraLite LOCATION 函数, 372
- 字符串文字
  - UltraLite 常量, 304
- 字符串运算符
  - UltraLite 动态 SQL 语法, 317
- 字符函数
  - UltraLite 按字母顺序排序的列表, 327
- 字符集
  - Palm OS 上的 UltraLite, 33
  - UltraLite char\_set 属性, 195
  - UltraLite 同步, 32
  - UltraLite 字符串, 282
  - UltraLite 归类创建参数, 170
  - UltraLite 数据库, 33
  - Windows Mobile 上的 UltraLite, 33
  - Windows 上的 UltraLite, 33
  - 术语定义, 508
- 字符集注意事项
  - UltraLite, 32
- 最大
  - UltraLite 物理限制, 6
- 最大值
  - UltraLite 日期范围, 290
- 最接近世纪值转换注意事项
  - UltraLite 关于, 179
- 最小值
  - UltraLite 日期范围, 290

---