



SQL Anywhere® 服务器 数据库管理

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

关于本书	xii
关于 SQL Anywhere 文档	xiii
启动并连接到数据库	1
教程：使用示例数据库	3
第 1 课：制作一个示例数据库副本	4
第 2 课：启动 SQL Anywhere 数据库服务器	5
第 3 课：显示数据库服务器消息窗口	6
第 4 课：停止数据库服务器	8
小结	9
使用数据库文件	11
数据库文件概述	12
预定义 dbspace	13
事务日志	14
创建数据库	20
使用附加 dbspace	23
使用实用程序数据库	28
消除数据库	32
运行数据库服务器	35
运行 SQL Anywhere 数据库服务器简介	36
启动数据库服务器	40
一些常见的选项	43
停止数据库服务器	53
启动和停止数据库	54
在当前会话外部运行服务器	57
对服务器启动进行故障排除	68
运行经过验证的 SQL Anywhere 应用程序	70
运行 SQL Anywhere Web 版应用程序	75
SQL Anywhere 中的错误报告	76
SQL Anywhere 数据库连接	79
连接参数	80
通过 SQL Anywhere API 连接	83

从桌面应用程序连接到 Windows Mobile 数据库	85
通过 Sybase Central、Interactive SQL 或 SQL Anywhere 控制台实用程序进行 连接	86
创建 ODBC 数据源	90
使用 OLE DB 连接到数据库	97
使用集成登录	99
Kerberos 验证	107
示例 SQL Anywhere 数据库连接	116
对连接进行故障排除	124
从数据库断开连接	132
客户端/服务器通信	133
支持的网络协议	134
使用 TCP/IP 协议	135
调整通信压缩设置以改善性能	141
网络通信故障排除	143
数据库服务器	147
SQL Anywhere 数据库服务器	148
数据库服务器选项	156
数据库选项	234
连接参数和网络协议选项	247
连接参数	248
网络协议选项	284
SQL Anywhere for Windows Mobile	309
在 Windows Mobile 设备上安装 SQL Anywhere	310
使用 Windows Mobile 示例应用程序	313
连接到在 Windows Mobile 设备上运行的数据库	318
配置 Windows Mobile 数据库	321
在 Windows Mobile 上运行数据库服务器	329
在 Windows Mobile 上使用管理实用程序	330
Windows Mobile 上的 SQL Anywhere 功能支持	336
配置数据库	341
SQL Anywhere 环境变量	343
SQL Anywhere 环境变量简介	344
DYLD_LIBRARY_PATH 环境变量 [Mac OS X]	346

LD_LIBRARY_PATH 环境变量 [Linux 和 Solaris]	347
LIBPATH 环境变量 [AIX]	348
ODBCHOME 环境变量 [Unix]	349
ODBCINI 和 ODBC_INI 环境变量 [Unix]	350
PATH 环境变量	351
SACHARSET 环境变量	352
SADIAGDIR 环境变量	353
SALANG 环境变量	355
SALOGDIR 环境变量	356
SATMP 环境变量	357
SHLIB_PATH 环境变量 [HP-UX]	359
SQLANY11 环境变量	360
SQLANYSAMP11 环境变量	361
SQLCONNECT 环境变量	362
SQLPATH 环境变量	363
SQLREMOTE 环境变量	364
SYBASE 环境变量	365
TMP、TMPDIR 和 TEMP 环境变量	366
文件位置和安装设置	367
安装目录结构	368
SQL Anywhere 如何定位文件	370
注册表和 INI 文件	373
国际语言和字符集	377
SQL Anywhere 的本地化版本	378
了解字符集	384
了解区域设置	389
了解归类	392
有关国际语言和字符集的任务	399
字符集和归类参考信息	404
管理用户 ID、特权和权限	415
管理登录策略概述	416
数据库权限和特权概述	422
管理用户权限和特权概述	430
管理连接的用户	442
管理组	443
数据库对象名和前缀	450

使用视图和过程来获得更高的安全性	452
更改嵌套对象的所有权	454
如何确定用户权限	456
管理连接所使用的资源	457
目录中的用户和权限	458
数据库选项	461
数据库选项简介	462
连接属性、数据库属性和数据库服务器属性	565
连接属性	566
数据库服务器属性	588
数据库属性	601
物理限制	613
SQL Anywhere 大小和数量限制	614
管理数据库	617
SQL Anywhere 图形管理工具	619
使用 Sybase Central	620
使用 Interactive SQL	636
使用文本完成	683
使用快速启动程序选项	686
使用 SQL Anywhere 控制台实用程序	687
检查软件更新	690
数据库管理实用程序	693
管理实用程序概述	695
备份实用程序 (dbbackup)	698
广播转发器实用程序 (dbns11)	703
证书创建实用程序 (createcert)	705
证书查看器实用程序 (viewcert)	708
数据源实用程序 (dbdsn)	710
dbisqlc 实用程序 (不建议使用)	720
消除实用程序 (dberase)	722
文件隐藏实用程序 (dbfhide)	724
直方图实用程序 (dbhist)	726
信息实用程序 (dbinfo)	728
初始化实用程序 (dbinit)	729

Interactive SQL 实用程序 (dbisql)	740
密钥对生成器实用程序 (createkey)	744
语言选择实用程序 (dblang)	745
日志传送管理器实用程序 (dbltm)	748
翻译日志文件实用程序 (dbtran)	753
Ping 实用程序 (dbping)	757
重建实用程序 (rebuild)	760
脚本执行实用程序 (dbrunsql)	761
服务器枚举实用程序 (dblocate)	763
服务器授权实用程序 (dblic)	766
用于 Linux 的服务实用程序 (dbsvc)	769
用于 Windows 的服务实用程序 (dbsvc)	773
SQL Anywhere 控制台实用程序 (dbconsole)	780
后台启动服务器实用程序 (dbspawn)	781
停止服务器实用程序 (dbstop)	783
支持实用程序 (dbsupport)	785
事务日志实用程序 (dblog)	792
卸载实用程序 (dbunload)	795
升级实用程序 (dbupgrad)	808
校验实用程序 (dbvalid)	810
版本诊断实用程序 (dbversion)	813
维护数据库	815
备份和数据恢复	817
备份快速入门	818
备份类型	819
选择备份格式	823
备份和恢复限制	824
创建服务器端备份	825
创建客户端备份	830
校验备份	832
恢复数据库	833
设计备份和恢复计划	843
备份参与同步和复制的数据库	847

内部备份过程	852
校验数据库	857
校验简介	858
使用校验和检测损坏	859
在校验数据库时提高性能	862
使用调度和事件自动完成任务	863
调度与事件使用简介	864
了解事件	865
了解调度	866
了解系统事件	868
了解事件处理程序	871
调度和事件内部	873
事件处理任务	875
SQL Anywhere 高可用性	879
数据库镜像简介	880
教程：使用数据库镜像	887
教程：使用多个数据库共享一个仲裁服务器的数据库镜像	891
设置数据库镜像	896
使用 SQL Anywhere Veritas 聚簇服务器代理	905
监控数据库	911
SQL Anywhere 监控器	913
SQL Anywhere 监控器简介	914
监控器快速入门	917
教程：使用监控器	918
启动监控器	923
退出监控器	924
连接到监控器	925
断开与监控器的连接	926
监控资源	927
管理资源	935
使用监控器用户	942
警告	946
安装的对象	950
在一台单独的计算机上安装 SQL Anywhere 监控器	951

监控器疑难解答	952
SQL Anywhere SNMP Extension Agent	955
SQL Anywhere SNMP Extension Agent 简介	956
了解 SNMP	957
使用 SQL Anywhere SNMP Extension Agent	961
SQL Anywhere MIB 参考	969
RDBMS MIB 参考	995
安全	1001
保护数据的安全	1003
安全功能简介	1004
安全提示	1005
控制数据库访问	1006
审计数据库活动	1012
以安全的方式运行数据库服务器	1018
加密和解密数据库	1019
保护 Windows Mobile 数据库的安全	1029
传送层安全	1031
传送层安全简介	1032
设置传送层安全	1035
创建数字证书	1036
加密 SQL Anywhere 客户端/服务器通信	1042
加密 SQL Anywhere Web 服务	1046
加密 MobiLink 客户端/服务器通信	1047
证书实用程序	1054
复制	1055
将 SQL Anywhere 用作 Open Server	1057
Open Client、Open Server 和 TDS	1058
将 SQL Anywhere 设置为 Open Server	1060
配置 Open Server	1062
Open Client 和 jConnect 连接的特性	1067
使用复制服务器复制数据	1069
将 SQL Anywhere 与复制服务器一起使用简介	1070

教程：使用复制服务器复制数据	1073
为复制服务器配置数据库	1081
使用 LTM	1084
术语表	1093
术语表	1095
索引	1123

关于本书

主题

本书介绍如何运行、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。

读者

本书适用于所有 SQL Anywhere 用户。您可以将它与文档集中的其它书籍结合使用。

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*”一节第 360 页。

- **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 SQLANYAMP11，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“SQLANYAMP11 环境变量”一节第 361 页。

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。

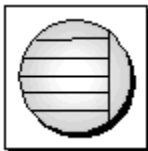
图标

本文档中使用了下列图标。

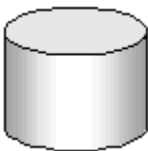
- 客户端应用程序。



- 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。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。

启动并连接到数据库

本节介绍如何启动 SQL Anywhere 数据库服务器以及如何从客户端应用程序连接到您的数据库。

教程：使用示例数据库	3
使用数据库文件	11
运行数据库服务器	35
SQL Anywhere 数据库连接	79
客户端/服务器通信	133
数据库服务器	147
连接参数和网络协议选项	247
SQL Anywhere for Windows Mobile	309

教程：使用示例数据库

目录

第 1 课：制作一个示例数据库副本	4
第 2 课：启动 SQL Anywhere 数据库服务器	5
第 3 课：显示数据库服务器消息窗口	6
第 4 课：停止数据库服务器	8
小结	9

第 1 课：制作一个示例数据库副本

本教程着重介绍示例数据库。该示例数据库表示一家小型公司，它生产相当有限的几种运动服装。其中包含公司的内部信息（雇员、部门和财务数据）以及产品信息（产品）和销售信息（销售订单、客户和联系方式）。该示例数据库中的所有信息都是虚构的。请参见“[关于示例数据库](#)”一节《[SQL Anywhere 11 - 简介](#)》。

在开始之前，请先制作示例数据库的副本，以便在修改数据库后能够恢复该数据库。

◆ 复制示例数据库

1. 创建一个目录以存放将在本教程中使用的示例数据库的副本，例如 `c:\demodb`。
2. 将示例数据库从 `samples-dir\demo.db` 复制到 `c:\demodb`。

有关 `samples-dir` 的信息，请参见“[示例目录](#)”一节第 368 页。

第 2 课：启动 SQL Anywhere 数据库服务器

◆ 启动运行示例数据库的个人数据库服务器（命令提示符）

- 运行以下命令启动个人数据库服务器，使用 `-n` 服务器选项将此服务器命名为 `mydemo11`，并连接到示例数据库的副本：

```
dbeng11 -n mydemo11 c:\demodb\demo.db
```

在 Windows 上，该数据库服务器会作为一个图标显示在系统任务栏中。

有关启动网络数据库服务器的详细信息，请参见“[连接到网络上的服务器](#)”一节第 120 页。

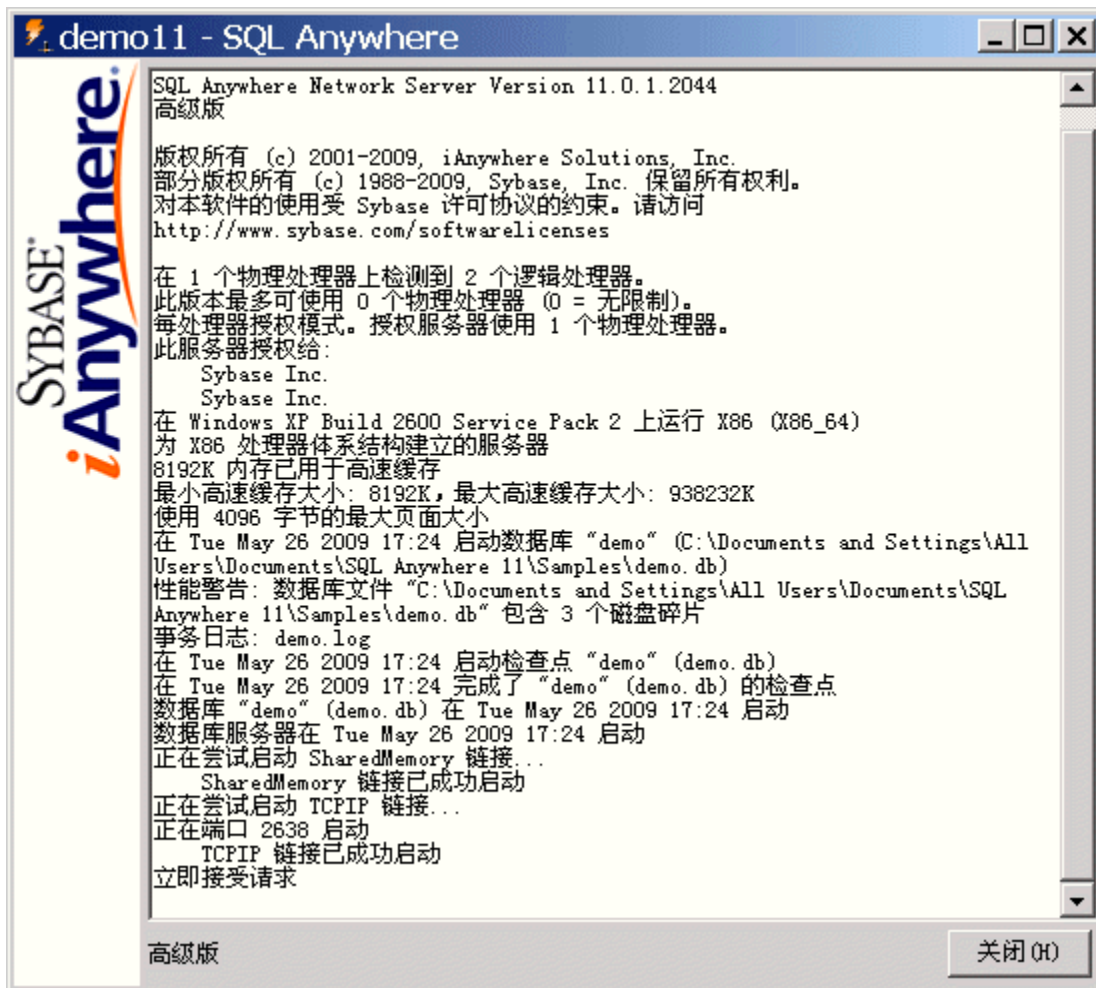
另请参见

- “[运行数据库服务器](#)” 第 35 页
- “[数据库服务器](#)” 第 147 页

第 3 课：显示数据库服务器消息窗口

您已经成功地启动了运行示例数据库的个人数据库服务器。然而，您还不能查看和处理数据库中的数据。

SQL Anywhere 个人服务器图标是表示有变化发生的唯一可见指示。您可以通过双击系统任务栏中的 SQL Anywhere 个人服务器图标在 Windows 中显示数据库服务器消息窗口。



数据库服务器消息窗口显示许多有用的信息，包括：

- **服务器名** 标题栏中的名称（在本例中为 mydemo11）是**服务器名**。在本教程中，使用 -n 服务器选项来指派服务器名。如果您不提供服务器名，则数据库服务器被赋予所启动的第一个数据库的名称。在应用程序连接到数据库时，可以使用此名称。请参见“命名服务器和数据库”一节第 43 页。
- **版本号 and 内部版本号** 服务器名后面的数字（例如 11.0.0.1083）是版本号和内部版本号。版本号表示 SQL Anywhere 的具体版本，内部版本号则与软件被编译的具体实例相关。

- **启动信息** 在数据库服务器启动时，它会留出一些内存，以便在处理数据库请求时使用。此保留的内存称为**高速缓存**。窗口中将显示高速缓存的大小。高速缓存以固定大小的**页**的形式进行编排，窗口中还会显示页大小。
- **数据库信息** 在窗口中显示数据库文件及其事务日志文件的名称。

在本例中，启动高速缓存大小和页大小都是缺省值。在大多数情况下（包括在本教程中），使用缺省启动选项即可。

第 4 课：停止数据库服务器

现在，可以停止您刚才启动的数据库服务器。

在 Windows 中，您可以通过单击数据库服务器消息窗口中的 [关闭] 来停止数据库服务器。

◆ 停止运行示例数据库的数据库服务器 (Windows)

1. 双击系统任务栏中的 SQL Anywhere 图标。
2. 单击 [关闭]。

◆ 停止运行示例数据库的数据库服务器（命令提示符）

- 运行以下命令停止运行示例数据库的个人数据库服务器：

```
dbstop mydemo11
```

停止服务器实用程序 (dbstop) 只能在命令行提示符环境中运行。请参见“[停止服务器实用程序 \(dbstop\)](#)”一节第 783 页。

教程清理

关闭数据库服务器后，可以删除 `c:\demodb` 目录及其内容。

小结

在本教程中，您学习了如何制作示例数据库的副本、如何启动运行示例数据库的数据库服务器以及如何查看数据库服务器消息窗口的内容。此外，还学习了如何停止数据库服务器。

另请参见

- “启动 Interactive SQL” 一节第 637 页
- “通过 Sybase Central、Interactive SQL 或 SQL Anywhere 控制台实用程序进行连接” 一节第 86 页
- “运行数据库服务器” 第 35 页

使用数据库文件

目录

数据库文件概述	12
预定义 dbspace	13
事务日志	14
创建数据库	20
使用附加 dbspace	23
使用实用程序数据库	28
消除数据库	32

数据库文件概述

每个数据库都具有与其关联的以下文件：

- **数据库文件** 该文件保存数据库信息。它的扩展名通常为 *.db*。
- **事务日志** 该文件保存对数据库进行的更改的记录，并且是恢复和同步所必需的。它的扩展名通常为 *.log*。请参见“[事务日志](#)”一节第 14 页。
- **临时文件** 数据库服务器使用临时文件保存数据库会话过程中所需的信息。数据库服务器在数据库关闭之后放弃该文件—即使该服务器仍在运行。该文件的名称由服务器生成，其扩展名为 *.tmp*。

临时文件的位置可在使用 **-dt** 服务器选项启动数据库服务器时指定。如果在启动数据库服务器时未指定临时文件的位置，则按顺序检查以下环境变量：

- SATMP 环境变量
- TMP 环境变量
- TMPDIR 环境变量
- TEMP 环境变量

如果这些环境变量均未定义，则在 Windows 操作系统中，SQL Anywhere 会将其临时文件放在当前目录中，而在 Unix 中，则会放在 */tmp* 目录中。

数据库服务器创建、维护和删除临时文件。您只需确保有足够的可用空间用于临时文件。您可以使用 `sa_disk_free_space` 过程来获取有关临时文件的可用空间信息。请参见“[sa_disk_free_space 系统过程](#)”一节《[SQL Anywhere 服务器 - SQL 参考](#)》。

- **预定义 dbspace 文件** 这些是存储用户数据的文件以及数据库使用的其它文件。请参见“[预定义 dbspace](#)”一节第 13 页。

附加文件

其它文件也可以成为数据库系统的一部分，其中包括：

- **Dbspace 文件** 除数据库文件外，您还可以将数据分散到几个独立的文件中。请参见“[CREATE DBSPACE 语句](#)”一节《[SQL Anywhere 服务器 - SQL 参考](#)》。

有关 dbspace 的信息，请参见“[使用附加 dbspace](#)”一节第 23 页。

- **事务日志镜像文件** 为了更加安全起见，您可以为事务日志创建镜像副本。该文件的扩展名通常为 *.mlg*。请参见“[事务日志镜像](#)”一节第 15 页。

预定义 dbspace

SQL Anywhere 将以下预定义 dbspace 用于数据库：

Dbspace	名称
主数据库文件	system
临时文件	temporary 或 temp
事务日志文件	translog
事务日志镜像	translogmirror

您不能使用这些名称创建用户定义 dbspace，也不能删除预定义 dbspace。

如果您通过使用预定义 dbspace 名称的用户定义 dbspace 升级版本 10.0.0 或更低版本数据库，则假定在 SQL 语句中对这些 dbspace 的所有引用都是引用用户定义 dbspace，而不是引用预定义 dbspace。可以引用预定义 dbspace 的唯一方式是通过删除用户定义 dbspace 或重命名它们，以便不使用与预定义 dbspace 相同的名称。

ALTER DBSPACE 语句支持预定义 dbspace 名称，所以可以向其中添加更多空间。请参见“ALTER DBSPACE 语句”一节《SQL Anywhere 服务器 - SQL 参考》。

DB_EXTENDED_PROPERTY 函数也接受预定义 dbspace 名称。请参见“DB_EXTENDED_PROPERTY 函数 [System]”一节《SQL Anywhere 服务器 - SQL 参考》。

事务日志

事务日志是一个独立于数据库文件的文件。它存储对数据库进行的所有更改，所有的插入、更新、删除、提交、回退和数据库模式变化都会被记录。事务日志还称作**前滚日志**或**重做日志**。

事务日志是备份和恢复的重要组成部分，也是使用 MobiLink 进行数据同步、使用 SQL Remote 或复制代理复制数据或进行数据库镜像所必需的。

缺省情况下，所有数据库都使用事务日志。事务日志的使用是可选的，但是，除非您因特殊原因而不使用，否则您应始终使用它。运行带有事务日志的数据库可提供更强大的故障保护功能、更好的性能以及数据复制功能。

建议您在计算机的不同磁盘上分别存储数据库文件和事务日志。如果 dbspace 和事务日志位于同一磁盘上，那么出现磁盘故障时，将丢失所有数据。但是，如果数据库和事务日志存储在不同的磁盘上，则在发生磁盘故障时，即使不能恢复全部数据，也能恢复大部分数据，因为在完整的数据库或事务日志（从中可以恢复数据库）这两者中，总有一个存在。

请参见“防止出现介质故障”一节第 846 页。

小心

数据库文件与事务日志文件必须与数据库服务器位于同一台物理计算机上，或者必须通过 SAN 或 iSCSI 配置进行访问。如果数据库文件和事务日志文件位于远程网络目录上，则可能会导致性能很差、数据损坏和服务器不稳定。

有关详细信息，请访问 <http://www.sybase.com/detail?id=1034790>。

何时将更改强制保存到磁盘上

同数据库文件一样，事务日志编排为**页**，即：内存中大小固定的区域。当在事务日志中记录更改时，是针对内存中的页进行的。在发生下列操作之前，就将所进行的更改强制保存到磁盘上：

- 页已满。
- 执行 COMMIT。

确保将已完成的事务存储在磁盘上，同时通过避免在每次操作时都写入磁盘而使性能得到改善。

利用配置选项可允许高级用户调整事务日志的精确行为。请参见“[cooperative_commits 选项 \[数据库\]](#)”一节第 494 页和“[delayed_commits 选项 \[数据库\]](#)”一节第 501 页。

另请参见

- “控制事务日志大小”一节第 16 页
- “-m 服务器选项”一节第 194 页
- “-m 数据库选项”一节第 238 页
- “sa_disk_free_space 系统过程”一节 《SQL Anywhere 服务器 - SQL 参考》
- “delete_old_logs 选项 [MobiLink 客户端] [SQL Remote] [复制代理]”一节第 502 页

事务日志镜像

事务日志镜像是事务日志的一个完全相同的副本，与事务日志同时接受维护。如果数据库有事务日志镜像，则每个数据库更改既写入到事务日志也写入事务日志镜像。缺省情况下，数据库没有事务日志镜像。

事务日志镜像为重要数据提供额外保护。它在事务日志出现介质故障时进行完整的数据恢复。事务日志镜像还使数据库服务器能够在启动数据库时自动执行事务日志校验。

建议您在运行大容量或重要应用程序时使用事务日志镜像。例如，在 SQL Remote 设置的统一数据库中，复制依赖事务日志，如果事务日志受损或被破坏，数据复制可能会失败。

使用事务日志镜像时，如果在尝试写入某个日志时出错（例如，如果磁盘已满），则数据库服务器会停止。事务日志镜像的作用是确保当任一日志设备出现介质故障时能够完全恢复；如果服务器继续使用单个事务日志，则镜像功能不再起作用了。

您可以在启动数据库服务器时指定 `-fc` 选项，以在数据库服务器出现文件系统已满时执行回调函数。请参见“[-fc 服务器选项](#)”一节第 175 页。

在何处存储事务日志镜像

因为每个数据库日志写操作都必须执行两次，所以使用事务日志镜像会有性能损失。性能损失取决于数据库通信的性质和通信量以及数据库和日志的物理配置。

事务日志镜像与事务日志应分别保存在不同的设备上。这样做可改善性能，而且如果任一设备出现故障，日志的另一个副本可保持数据的安全以进行恢复。

事务日志镜像的替代方法

除了事务日志镜像之外，还可以使用以下配置作为替代方法：

- 数据库镜像。请参见“[数据库镜像简介](#)”一节第 880 页。
- 提供硬件镜像的磁盘控制器。通常，硬件镜像要比操作系统级软件镜像成本更高，但提供的性能更好。
- 操作系统级软件镜像，如由 Microsoft Windows 提供的镜像。

活动备份可提供额外保护，这与使用事务日志镜像有某些相似之处。请参见“[活动备份和事务日志镜像之间的区别](#)”一节第 822 页。

有关创建具有事务日志镜像的数据库的信息，请参见“[初始化实用程序 \(dbinit\)](#)”一节第 729 页。

有关更改现有数据库以使用事务日志镜像的信息，请参见“[事务日志实用程序 \(dblog\)](#)”一节第 792 页。

更改事务日志的位置

在更改事务日志的位置时，不能运行数据库。

有关如何选择事务日志位置的详细信息，请参见“[事务日志](#)”一节第 14 页。

◆ **更改事务日志的位置 (Sybase Central):**

1. 从 [工具] 菜单选择 [SQL Anywhere 11] » [更改日志文件设置]。
2. 按照 [更改日志文件设置向导] 中的说明进行操作。

◆ **更改现有数据库的事务日志镜像的位置 (命令行)**

1. 确保数据库没有运行。
2. 运行以下命令:

```
dblog -t new-transaction-log-file database-file
```

另请参见

- [“事务日志实用程序 \(dblog\)” 一节第 792 页](#)

启动现有数据库的事务日志镜像

使用事务日志实用程序，可以在数据库未运行时维护现有数据库的事务日志镜像。

◆ **启动现有数据库的事务日志镜像 (Sybase Central):**

1. 从 [工具] 菜单选择 [SQL Anywhere 11] » [更改日志文件设置]。
2. 按照 [更改日志文件设置向导] 中的说明进行操作。

◆ **启动现有数据库的事务日志镜像 (命令行)**

1. 确保数据库没有运行。
2. 运行以下命令:

```
dblog -m mirror-file database-file
```

您还可以使用 dblog 实用程序和 Sybase Central 禁止数据库使用事务日志镜像。

另请参见

- [“事务日志实用程序 \(dblog\)” 一节第 792 页](#)

控制事务日志大小

事务日志的大小也会影响恢复时间。您可以通过确保所有表都有一个精简的主键来控制事务日志文件的生长。如果执行更新或删除的表没有不允许为 NULL 的主键或唯一索引，则受影响的行的全部内容都将输入到事务日志中。如果定义了主键，则数据库服务器只需存储主键列的值以便唯一标识行。如果该表包含许多列或宽列，若没有定义主键，则事务日志页会更快被填满。除了占用磁盘空间以外，这个额外的数据写操作也影响性能。

如果没有主键，服务器会在表上查找 UNIQUE NOT NULL 索引（或 UNIQUE 约束）。只有允许 NULL 的 UNIQUE 索引是不够的。

另请参见

- “-m 服务器选项”一节第 194 页
- “-m 数据库选项”一节第 238 页
- “sa_disk_free_space 系统过程”一节 《SQL Anywhere 服务器 - SQL 参考》
- “delete_old_logs 选项 [MobiLink 客户端] [SQL Remote] [复制代理]”一节第 502 页

确定哪个连接有未完成的事务

如果执行重命名或删除事务日志的备份，则未完成的事务将转入新的事务日志中。

您可以使用系统过程来确定哪个用户有未完成的事务。如果连接数量不太多，您还可以使用 SQL Anywhere 控制台实用程序来确定哪个连接有无完成的事务。如有必要，您可以用 DROP CONNECTION 语句断开用户的连接。

◆ 确定哪个连接有未完成的事务 (SQL):

1. 从 Interactive SQL 连接到数据库。
2. 执行 sa_conn_info 系统过程:

```
CALL sa_conn_info;
```

3. 检查 UncommitOps 列以查看哪个连接有无提交的操作。

请参见“sa_conn_info 系统过程”一节 《SQL Anywhere 服务器 - SQL 参考》。

◆ 确定哪个连接有未完成的事务 (SQL Anywhere 控制台实用程序):

1. 从 SQL Anywhere 控制台实用程序连接到数据库。

例如，以下命令使用用户 ID DBA 和口令 sql 连接到缺省数据库:

```
dbconsole -c "UID=DBA;PWD=sql"
```

请参见“SQL Anywhere 控制台实用程序 (dbconsole)”一节第 780 页。

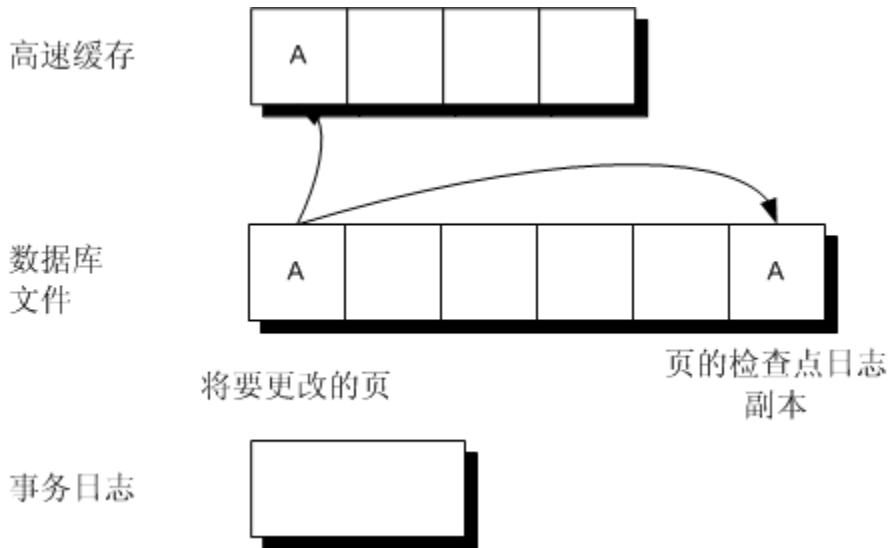
2. 双击每个连接，检查 [未提交的操作] 条目以查看哪些用户有无提交的操作。如有必要，您可以断开与用户的连接以使备份能够完成。

了解检查点日志

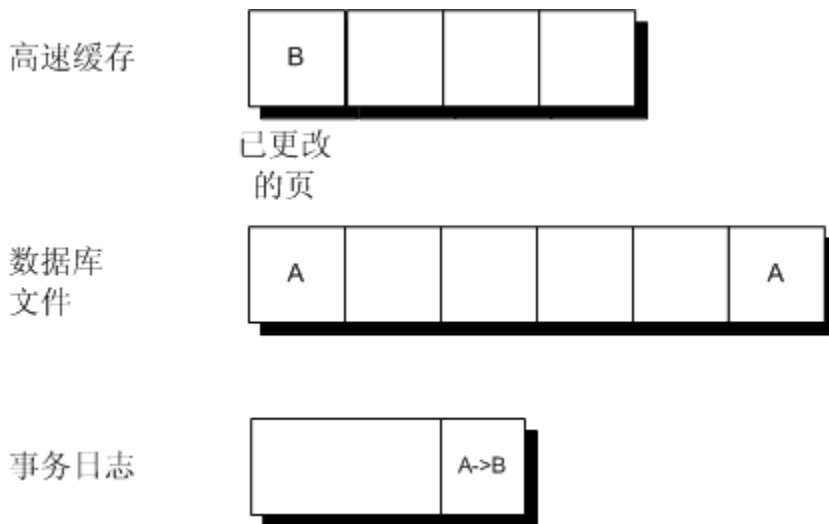
数据库文件由页（即硬盘中大小固定的区域）组成。**检查点日志**位于数据库文件的末尾处，并存储在系统 dbspace 中。在会话过程中，页会根据需要添加到检查点日志中，整个检查点日志会在会话结束时删除。

在任何页更新（弄脏）之前，数据库服务器都执行以下操作:

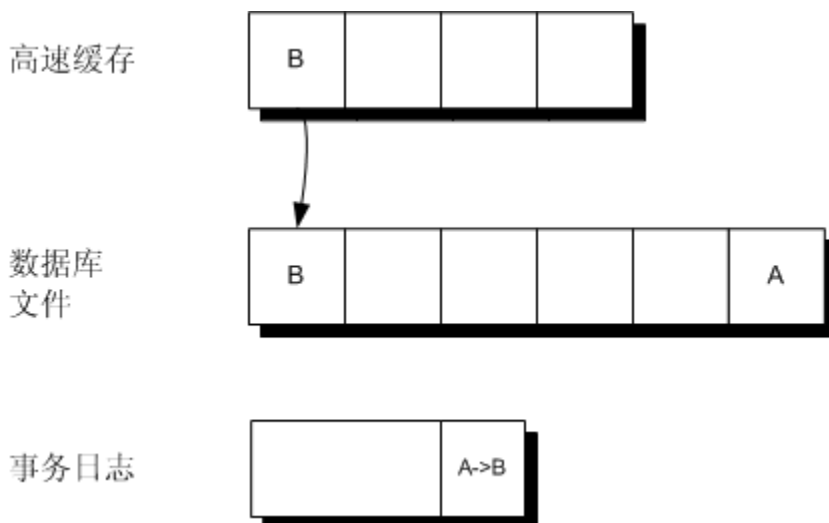
- 它将该页读入内存的数据库高速缓存中。
- 它创建原始页的副本。这些复制的页称为检查点日志。



对页进行的更改应用于高速缓存中的副本。出于性能上的考虑，它们不立即写入磁盘上的数据库文件中。



当高速缓存变满时，已更改的页可写入到磁盘中。检查点日志中的副本保持不变。



了解检查点

检查点是一个点，在到达该点时，所有脏页都写入到磁盘中，因而该点代表磁盘中数据库的一种已知的一致状态。在检查点之后，检查点日志的内容都将删除。空检查点日志页在给定会话内保留在检查点日志中，并且可重新用于新的检查点日志数据。随着检查点日志逐渐增大，数据库文件也逐渐增大。

在检查点，数据库中的所有数据都保存在磁盘上的数据库文件中。数据库文件中的信息与事务日志中的信息相匹配。在恢复过程中，数据库首先恢复到最近的检查点，然后应用自该检查点以来进行的更改。

整个检查点日志（包括所有空的检查点日志页）会在每个会话结束时删除。删除检查点日志将使数据库缩小。

数据库服务器可以启动检查点并在出现检查点时执行其它操作。不过，如果检查点操作已经在进行，则启动新检查点的任何操作（如 ALTER TABLE 或 CREATE INDEX）必须等到当前检查点操作结束之后方可执行。

另请参见

- “备份和恢复限制”一节第 824 页
- “了解备份”一节第 852 页
- “数据库服务器如何确定何时进行检查点操作”一节第 853 页

创建数据库

您可使用 Sybase Central、Interactive SQL 或命令行来创建或初始化 SQL Anywhere 数据库。创建数据库之后，可以连接到该数据库并构建各表和其它对象。

其它应用程序设计系统（例如 Sybase PowerDesigner Physical Data Model）也包含用于创建数据库对象的工具。这些工具会构建通常通过其 ODBC 接口提交给数据库服务器的 SQL 语句。如果您在使用这些工具中的某一个工具，则不需要构建 SQL 语句就可以完成诸如创建表、指派权限等操作。请参见“关于 PowerDesigner Physical Data Model”一节《SQL Anywhere 11 - 简介》。

有关数据库设计的详细信息，请参见“在 SQL Anywhere 中创建数据库”《SQL Anywhere 服务器 - SQL 的用法》。

事务日志

创建数据库时，必须确定放置事务日志的位置。此日志按照进行更改的时间顺序存储对数据库所做的所有更改。当数据库文件上出现介质故障时，事务日志对进行数据库恢复至关重要。它还可以使您的工作更加高效。缺省情况下，事务日志与数据库文件放在相同的目录中，但建议在生产中不要这样做。

有关放置事务日志的详细信息，请参见“事务日志”一节第 14 页。

数据库文件兼容性

SQL Anywhere 数据库是一个操作系统文件，可以复制到其它位置，复制方法与任何其它文件都一样。

除了文件系统存在文件大小限制或 SQL Anywhere 支持大文件的情况外，数据库文件可以在所有操作系统间兼容。请参见“SQL Anywhere 大小和数量限制”一节第 614 页。

通过复制数据库文件的方式，在任何操作系统中所创建的数据库都可以在另一个操作系统中使用。同样，通过个人数据库服务器创建的数据库也可用于网络服务器。SQL Anywhere 数据库服务器可以管理由软件的早期版本所创建的数据库，但旧服务器不能管理较新的数据库。

创建数据库 (Sybase Central)

可以在 Sybase Central 中使用 [创建数据库向导] 创建数据库。请参见“创建数据库 (SQL)”一节第 21 页和“创建数据库 (命令行)”一节第 22 页。

◆ 创建新数据库 (Sybase Central)

1. 启动 Sybase Central。
2. 选择 [工具] » [SQL Anywhere 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。

提示

您还可以使用以下方法从 Sybase Central 中访问 [\[创建数据库向导\]](#)。

- 选择服务器，然后选择 [\[文件\]](#) » [\[创建数据库\]](#)。
- 右击服务器，然后选择 [\[创建数据库\]](#)。

创建 Windows Mobile 数据库

有关创建 Windows Mobile 数据库的信息，请参见 [“创建 Windows Mobile 数据库”](#) 一节第 322 页。

创建数据库 (SQL)

在 Interactive SQL 中，使用 CREATE DATABASE 语句创建数据库。使用此语句之前，需要连接到一个现有的数据库。

◆ 创建新数据库 (SQL)

1. 启动名为 sample 的数据库服务器。

```
dbeng11 -n sample
```

2. 启动 Interactive SQL。
3. 连接到一个现有的数据库。如果没有数据库，可以连接到实用程序数据库 utility_db。请参见 [“连接到实用程序数据库”](#) 一节第 29 页。
4. 执行 CREATE DATABASE 语句。

请参见 [“CREATE DATABASE 语句”](#) 一节 《SQL Anywhere 服务器 - SQL 参考》。

示例

在 `c:\temp` 目录中使用文件名 `temp.db` 创建数据库文件。

```
CREATE DATABASE 'c:\\temp\\temp.db';
```

该目录路径是相对于数据库服务器的路径。执行此语句所需的权限是在服务器命令行上用 `-gu` 选项来设置的。缺省设置需要 DBA 权限。

反斜线在 SQL 中是转义字符，在某些情况下必须成对使用。`\x` 和 `\n` 序列可用于指定十六进制字符和换行符。`n` 和 `x` 以外的字母前面带有反斜线时，不具有任何特殊含义。下面是一些必须使用反斜线的示例。

```
CREATE DATABASE 'c:\\temp\\x41\x42\x43xyz.db';
```

开头的 `\\` 序列代表一个反斜线。`\x` 序列分别代表字符 A、B 和 C。此处的文件名是 `ABCxyz.db`。

```
CREATE DATABASE 'c:\temp\nest.db';
```

为避免 `\n` 序列被解释为换行符，使用了两个反斜线。

请参见 [“转义序列”](#) 一节 《SQL Anywhere 服务器 - SQL 参考》。

创建数据库（命令行）

可以从命令行使用初始化实用程序 (dbinit) 创建数据库。在此实用程序中，可以加入命令行选项来为数据库指定不同的设置。

◆ 创建新数据库（命令行）

- 运行 dbinit 命令。

例如，要创建 4 KB 页面大小的名为 *company.db* 的数据库，运行以下命令：

```
dbinit -p 4k company.db
```

另请参见

- “初始化实用程序 (dbinit)” 一节第 729 页

创建具有事务日志镜像的数据库

在创建数据库时，您可以选择维护事务日志镜像。该选项由 CREATE DATABASE 语句、Sybase Central 或 dbinit 实用程序提供。

有关为何需要使用事务日志镜像的详细信息，请参见“事务日志镜像”一节第 15 页。

◆ 创建使用事务日志镜像的数据库 (Sybase Central):

1. 从 [工具] 菜单选择 [SQL Anywhere 11] » [创建数据库]。
2. 请按照 [创建数据库向导] 中的说明进行操作。

◆ 创建使用事务日志镜像的数据库 (SQL):

- 使用包含 TRANSACTION LOG 和 MIRROR 子句的 CREATE DATABASE 语句。例如：

```
CREATE DATABASE 'c:\\mydb'  
TRANSACTION LOG ON mydb.log  
MIRROR 'd:\\mydb.mlg';
```

请参见“CREATE DATABASE 语句”一节《SQL Anywhere 服务器 - SQL 参考》。

◆ 创建使用事务日志镜像的数据库（命令行）

- 使用包含 -m 选项的 dbinit 实用程序。例如，以下命令（应当在一行上输入）初始化名为 *company.db* 的数据库，并将事务日志保留在另一设备上，而将镜像保留在第三个设备上。

```
dbinit -t d:\\log-dir\\company.log -m  
e:\\mirr-dir\\company.mlg c:\\db-dir\\company.db
```

请参见“初始化实用程序 (dbinit)”一节第 729 页。

使用附加 dbspace

通常是大数据库所需的

对于大多数数据库来说，单个数据库文件就已足够。但是，对于大数据库的用户来说，附加数据库文件有时是必需的。附加数据库文件还是将相关信息聚集在独立文件中的方便工具。

当您初始化某个数据库时，它包含一个数据库文件。这第一个数据库文件称为**主文件或系统 dbspace**。缺省情况下，所有数据库对象和所有数据都放在主文件中。

dbspace 是一个为数据创建更多空间的附加数据库文件。一个数据库可以包含在多达 13 个独立文件（一个主文件和 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。

临时表只在临时 **dbspace** 中创建。

有多种方法可指定于其中创建基表或其它数据库对象的 **dbspace**。在以下列表中，位置靠前的方法所指定的位置优先于位置靠后的方法所指定的位置。

1. IN DBSPACE 子句（如果已指定）
2. default_dbspace 选项（如果已设置）
3. 系统 dbspace

如果 **dbspace** 名称包含一个句点并且未加上引号，则数据库服务器为此名称生成一个错误。

每个数据库文件最大允许有 2^{28} （约 2 亿 6 千 8 百万）数据库页。例如，使用页面大小为 4 KB 的数据库创建的数据库文件最大可增至 1 TB ($2^{28} * 4$ KB)。但是，实际上，在其中创建文件的物理文件系统所允许的最大文件大小会对可允许的最大大小产生很大的影响。

尽管一些旧的文件系统将文件大小限制为最大不超过 2 GB，但是许多文件系统（如使用 NTFS 文件系统的 Windows 文件系统）允许利用完整的数据库文件大小。如果数据库中放置的数据量超过最大文件大小，就有必要将数据分成多个数据库文件。而且，您可能因大小限制以外的原因（例如聚簇相关对象）而想要创建多个 **dbspace**。

有关支持的操作系统所允许的最大文件大小的信息，请参见“[SQL Anywhere 大小和数量限制](#)”一节第 614 页。

您可以使用 `sa_disk_free` 系统过程获取有关 **dbspace** 可用空间的信息。请参见“[sa_disk_free_space 系统过程](#)”一节《[SQL Anywhere 服务器 - SQL 参考](#)》。

SYSDBSPACE 系统视图包含有关用于数据库的所有 **dbspace** 的信息。请参见“[SYSDBSPACE 系统视图](#)”一节《[SQL Anywhere 服务器 - SQL 参考](#)》。

拆分现有的数据库

如果您想要在多个 **dbspace** 之间拆分现有的数据库对象，则必须卸载数据库并修改所生成的命令文件（缺省情况下，名为 `reload.sql`），以便重建数据库。在 `reload.sql` 文件中，将 IN 子句添加到 CREATE TABLE 语句中，以便为不想放到主文件中的每一个表指定 **dbspace**。

对 dbspace 的权限

SQL Anywhere 支持对 dbspace 使用权限。只有 CREATE 权限受支持。CREATE 权限允许用户在指定的 dbspace 中创建数据库对象。可以通过执行 GRANT CREATE 语句授予 dbspace 的 CREATE 权限。请参见“GRANT 语句”一节《SQL Anywhere 服务器 - SQL 参考》。

Dbspace 权限具有如下性质：

- 试图用基础数据创建新对象的用户必须具有放置基础数据的 dbspace 的 CREATE 权限。
- 即使已发出 GRANT CREATE ON 语句，用户（被授予者）也必须具有 RESOURCE 权限才能创建新数据库对象。
- 放置于特定 dbspace 中以及需要 CREATE 权限的对象的当前列表包括表、索引、文本索引和实例化视图。请注意，诸如普通视图和过程的对象没有任何基础数据，因此不需要 CREATE 权限。
- 可以直接授予用户 CREATE 权限，也可以通过成为已授予此权限的组的成员继承此权限。
- 可以授予 PUBLIC 对特定 dbspace 的 CREATE 权限，在这种情况下，拥有 RESOURCE 权限的所有用户也可以在此 dbspace 上创建对象。
- 新创建的 dbspace 将其 CREATE 权限自动授予 PUBLIC。
- 可以撤销权限，例如，在试图保护 dbspace 安全时。也可以管理内部 dbspace（系统和临时）的权限以控制访问。
- 创建本地临时表不需要任何权限；dbspace 权限不影响创建本地临时表。但是，创建全局临时表需要对临时 dbspace 的 RESOURCE 权限和 CREATE 权限。

另请参见

- “CREATE DBSPACE 语句”一节《SQL Anywhere 服务器 - SQL 参考》
- “DB_EXTENDED_PROPERTY 函数 [System]”一节《SQL Anywhere 服务器 - SQL 参考》
- “CREATE TABLE 语句”一节《SQL Anywhere 服务器 - SQL 参考》
- “UNLOAD 语句”一节《SQL Anywhere 服务器 - SQL 参考》

创建 dbspace

您可以从 Sybase Central 或使用 CREATE DBSPACE 语句创建新数据库文件或 dbspace。新 dbspace 的数据库文件可与主文件位于相同的磁盘驱动器上，也可以位于另一个磁盘驱动器上。您必须拥有 DBA 权限才能创建 dbspace。

对于每个数据库，除了主 dbspace 以外，您还可以创建多达 12 个 dbspace。新建的 dbspace 是空的。当您创建新表或索引时，可利用 CREATE 语句中的 IN 子句将该表放在特定的 dbspace 中，也可在创建该表之前设置 default_dbspace 选项。如果不指定 IN 子句，也不更改 default_dbspace 选项的设置，则将在系统 dbspace 中创建该表。

每个表都完全包含在它创建时所在的 dbspace 中。缺省情况下，索引与它们的表出现在同一个 dbspace 中，但是可以通过将 IN 子句提供为 CREATE 语句的一部分，将它们放在独立的 dbspace 中。

另请参见

- “default_dbspace 选项 [数据库]” 一节第 499 页
- “CREATE DBSPACE 语句” 一节 《SQL Anywhere 服务器 - SQL 参考》
- “CREATE TABLE 语句” 一节 《SQL Anywhere 服务器 - SQL 参考》 “CREATE TABLE 语句” 一节 《SQL Anywhere 服务器 - SQL 参考》
- “CREATE INDEX 语句” 一节 《SQL Anywhere 服务器 - SQL 参考》

创建 dbspace**◆ 创建 dbspace (Sybase Central):**

1. 打开该数据库的 [Dbspace] 文件夹。
2. 选择 [文件] » [新建] » [Dbspace]。
3. 请按照 [创建 Dbspace 向导] 中的说明进行操作。

新 dbspace 将出现在 [Dbspace] 文件夹中。

◆ 创建 dbspace (SQL):

- 执行 CREATE DBSPACE 语句。

示例

以下命令在 *library.db* 文件（此文件与主文件位于同一个目录）中创建一个名为 MyLibrary 的新 dbspace:

```
CREATE DBSPACE MyLibrary
AS 'library.db';
```

以下命令创建 LibraryBooks 表，并将其放在 MyLibrary dbspace 中。

```
CREATE TABLE LibraryBooks (
title CHAR(100),
author CHAR(50),
isbn CHAR(30)
) IN MyLibrary;
```

以下命令将创建一个名为 MyLibrary 的新 dbspace，将缺省 dbspace 设置为 MyLibrary dbspace，然后在 MyLibrary dbspace 中创建 LibraryBooks 表。

```
CREATE DBSPACE MyLibrary
AS 'e:\dbfiles\library.db';
SET OPTION default_dbspace = 'MyLibrary';
CREATE TABLE LibraryBooks (
title CHAR(100),
author CHAR(50),
isbn CHAR(30),
);
```

另请参见

- “CREATE DBSPACE 语句”一节 《SQL Anywhere 服务器 - SQL 参考》
- “default_dbspace 选项 [数据库]”一节第 499 页
- “使用表”一节 《SQL Anywhere 服务器 - SQL 的用法》
- “CREATE INDEX 语句”一节 《SQL Anywhere 服务器 - SQL 参考》

为数据库文件预分配空间

创建新数据库文件时，可使用 CREATE DATABASE 语句的 DATABASE SIZE 子句或通过指定 dbinit -dbs 选项，来预分配数据库空间。请参见“CREATE DATABASE 语句”一节 《SQL Anywhere 服务器 - SQL 参考》和“初始化实用程序 (dbinit)”一节第 729 页。

使用数据库时，SQL Anywhere 将根据需要自动增大数据库文件。快速更改数据库文件可能会导致磁盘上有过多的文件碎片，从而导致潜在的性能问题。而且，多次小的分配要比一次大的分配慢。如果使用的是更改较为频繁的数据库，则可使用 Sybase Central 或 ALTER DBSPACE 语句为 dbspace 或事务日志预分配磁盘空间。

您必须具有 DBA 权限才能变更数据库文件的属性。

性能提示

在预分配磁盘空间之后运行磁盘碎片整理实用程序，有助于确保数据库文件不会分散在磁盘驱动器的多个不连续区域。如果数据库文件碎片过多，则可能影响性能。

◆ 预分配空间 (Sybase Central):

1. 打开 [Dbspace] 文件夹。
2. 右击 dbspace，然后选择 [预分配空间]。
3. 输入要添加至 dbspace 的空间量。您可以以页、字节、千字节 (KB)、兆字节 (MB)、千兆字节 (GB) 或千吉字节 (TB) 为单位添加空间。
4. 单击 [确定]。

◆ 预分配空间 (SQL):

1. 连接到数据库。
2. 执行 ALTER DBSPACE 语句。

示例

将系统 dbspace 的大小增加 200 页。

```
ALTER DBSPACE system  
ADD 200;
```

将系统 dbspace 的大小增加 400 兆字节。

```
ALTER DBSPACE system  
ADD 400 MB;
```

另请参见

- “创建 dbspace” 一节第 24 页
- “ALTER DBSPACE 语句” 一节 《SQL Anywhere 服务器 - SQL 参考》

删除 dbspace

您可以使用 Sybase Central 或 DROP DBSPACE 语句删除 dbspace。在删除某个 dbspace 之前，必须删除使用该 dbspace 的所有表和索引。您必须具有 DBA 权限才能删除 dbspace。

◆ 删除 dbspace (Sybase Central):

1. 打开 [Dbspace] 文件夹。
2. 右击 dbspace，然后选择 [删除]。

◆ 删除 dbspace (SQL):

1. 连接到数据库。
2. 执行 DROP DBSPACE 语句。

另请参见

- “删除表” 一节 《SQL Anywhere 服务器 - SQL 的用法》
- “DROP DBSPACE 语句” 一节 《SQL Anywhere 服务器 - SQL 参考》

使用实用程序数据库

实用程序数据库是一个物理上不存在的虚拟数据库。此功能使您不必先连接到现有物理数据库就可以执行数据库文件管理语句，如 CREATE DATABASE 语句。实用程序数据库没有数据库文件，因此它不能包含数据。

实用程序数据库命名为 **utility_db**。如果您试图用此名称创建或启动数据库，则操作将失败。

连接到实用程序数据库后执行以下语句将在 `c:\temp` 目录中创建名为 `new.db` 的数据库。

```
CREATE DATABASE 'c:\\temp\\new.db';
```

请参见“CREATE DATABASE 语句”一节《SQL Anywhere 服务器 - SQL 参考》。

您还可以使用实用程序数据库检索连接属性和服务器属性的值。

例如，如果针对实用程序数据库执行以下语句，则会返回将在创建数据库时使用的缺省归类序列：

```
SELECT PROPERTY( 'DefaultCollation' );
```

有关连接和数据库服务器属性的信息，请参见：

- “连接属性”一节第 566 页
- “数据库服务器属性”一节第 588 页

允许实用程序数据库使用的语句

当连接到实用程序数据库时，只能执行以下语句：

- ALTER DATABASE *dbfile* ALTER TRANSACTION LOG（请参见“ALTER DATABASE 语句”一节《SQL Anywhere 服务器 - SQL 参考》）
- “CREATE DATABASE 语句”一节《SQL Anywhere 服务器 - SQL 参考》
- “CREATE DECRYPTED DATABASE 语句”一节《SQL Anywhere 服务器 - SQL 参考》
- “CREATE DECRYPTED FILE 语句”一节《SQL Anywhere 服务器 - SQL 参考》
- “CREATE ENCRYPTED DATABASE 语句”一节《SQL Anywhere 服务器 - SQL 参考》
- “CREATE ENCRYPTED FILE 语句”一节《SQL Anywhere 服务器 - SQL 参考》
- “DROP DATABASE 语句”一节《SQL Anywhere 服务器 - SQL 参考》
- CREATE USER DBA IDENTIFIED BY *new-password*（请参见“CREATE USER 语句”一节《SQL Anywhere 服务器 - SQL 参考》）
- “RESTORE DATABASE 语句”一节《SQL Anywhere 服务器 - SQL 参考》
- REVOKE CONNECT FROM DBA（请参见“REVOKE 语句”一节《SQL Anywhere 服务器 - SQL 参考》）
- 不包含 FROM 或 WHERE 子句的 SELECT 语句（请参见“SELECT 语句”一节《SQL Anywhere 服务器 - SQL 参考》）
- “START DATABASE 语句”一节《SQL Anywhere 服务器 - SQL 参考》
- “STOP DATABASE 语句”一节《SQL Anywhere 服务器 - SQL 参考》
- “STOP ENGINE 语句”一节《SQL Anywhere 服务器 - SQL 参考》

连接到实用程序数据库

您可以通过在连接到数据库时将 `utility_db` 指定为数据库名，在数据库服务器上启动实用程序数据库。可使用 `-su` 服务器选项为 DBA 用户设置实用程序数据库口令，或禁用与实用程序数据库的连接。如果在启动实用程序数据库时并未指定 `-su` 选项，则个人服务器和网络服务器对于用户 ID 和口令的要求将会有所不同。

就个人数据库服务器而言，如果未指定 `-su`，则连接到实用程序数据库将不会有任何安全限制。对于个人服务器，必须指定用户 ID DBA。您还必须指定一个口令，但它可以是任何口令。假定能够连接到个人数据库服务器的任何人都可以直接访问文件系统，因此不根据口令筛选用户。

在使用 `-su` 选项时为避免以纯文本形式键入实用程序数据库口令，可创建一个包含该口令的文件，然后使用 `dbfhide` 实用程序将其进行模糊处理。例如，假设名为 `util_db_pwd.cfg` 的文件包含该实用程序数据库口令。可使用 `dbfhide` 对此文件进行模糊处理，然后将其重命名为 `util_db_pwd_hide.cfg`：

```
dbfhide util_db_pwd.cfg util_db_pwd_hide.cfg
```

然后可使用 `util_db_pwd_hide.cfg` 文件指定实用程序数据库口令：

```
dbsrv11 -su @util_db_pwd_hide.cfg -n my_server c:\mydb.db
```

请参见“文件隐藏实用程序 (`dbfhide`)”一节第 724 页。

对于网络服务器，如果未指定 `-su`，则必须指定用户 ID DBA，还必须指定存储在 `util_db.ini` 文件（与数据库服务器可执行文件存储在同一个目录中）中的口令。由于该目录位于服务器上，因此您可以控制对该文件的访问，从而控制谁能够访问该口令。此口令区分大小写。

注意

不建议使用 `util_db.ini` 文件。应使用 `-su` 服务器选项为实用程序数据库的 DBA 用户指定口令。请参见“`-su` 服务器选项”一节第 211 页。

◆ 连接到个人服务器上的实用程序数据库 (Interactive SQL)

1. 用以下命令启动数据库服务器：

```
dbeng11 -n TestEng
```

为了更加安全起见，可使用 `-su` 选项指定实用程序数据库口令。

2. 启动 Interactive SQL。
3. 在 [连接] 窗口中，在 [用户 ID] 中键入 DBA，然后键入任意非空口令。口令本身不被检查，但是此字段不得为空。
4. 在 [数据库] 选项卡上，输入 `utility_db` 作为 [数据库名]，输入 `TestEng` 作为 [服务器名]。
5. 单击 [确定] 进行连接。

Interactive SQL 将连接到个人服务器上名为 TestEng 的实用程序数据库。

◆ 连接到网络服务器上的实用程序数据库 (Interactive SQL)

1. 使用以下命令启动数据库服务器：

```
dbsrv11 -n TestEng -su 9Bx231K
```

2. 启动 Interactive SQL。
3. 在 [连接] 窗口中，键入 DBA 用作 [用户 ID]，然后键入通过 -su 选项指定的口令。
4. 在 [数据库] 选项卡上，输入 utility_db 作为 [数据库名]，输入 TestEng 作为 [服务器名]。
5. 单击 [确定] 进行连接。

Interactive SQL 将连接到网络服务器上名为 TestEng 的实用程序数据库。

请参见“SQL Anywhere 数据库连接”第 79 页和“-su 服务器选项”一节第 211 页。

注意

当连接到实用程序数据库时，如果执行 REVOKE CONNECT FROM DBA，则会禁用将来与实用程序数据库的连接。这意味着，除非您使用在执行 REVOKE CONNECT 之前便已存在的连接或重新启动数据库服务器，否则以后将无法连接到实用程序数据库。请参见“REVOKE 语句”一节《SQL Anywhere 服务器 - SQL 参考》。

对网络数据库服务器使用 util_db.ini（不建议使用）

注意

不建议使用 *util_db.ini* 文件。建议您使用 -su 服务器选项为实用程序数据库指定 DBA 用户的口令。

util_db.ini 的使用依赖于托管数据库服务器的计算机的物理安全性，因为 *util_db.ini* 文件可通过文本编辑器方便地读取。

就网络服务器而言，缺省情况下，如果不指定 -su 或不使用 *util_db.ini*，便无法连接到实用程序数据库。如果您使用 *util_db.ini*，则该文件将存储口令，与数据库服务器可执行文件位于同一目录，并包含以下文本：

```
[UTILITY_DB]  
PWD=password
```

要保护 *util_db.ini* 文件的内容不会受到偶然的直接访问，可使用文件隐藏实用程序 (dbfhide) 对文件进行简单加密。也可使用操作系统功能来限制对服务器文件系统的访问。

有关模糊处理 *.ini* 文件的详细信息，请参见“隐藏 *.ini* 文件的内容”一节第 724 页。

指定执行文件管理语句所需的权限

-gu 数据库服务器选项控制哪些人能够执行文件管理语句。可使用该选项指定哪些用户能够执行某些管理任务。请参见“-gu 服务器选项”一节第 188 页。

文件管理语句的使用权限共有以下四个级别：

-gu 选项	效果	适用于
all	任何用户都能够执行文件管理语句	任何数据库（包括实用程序数据库）
none	任何用户都不能执行文件管理语句	任何数据库（包括实用程序数据库）
DBA	只有具备 DBA 权限的用户才能执行文件管理语句	任何数据库（包括实用程序数据库）
utility_db	只有能够连接到实用程序数据库的用户才能执行文件管理语句	仅限实用程序数据库

示例

要防止文件管理语句被使用，请使用 `-gu` 选项的 `none` 权限级别启动数据库服务器。以下命令启动一个数据库服务器并将其命名为 `TestSrv`。它装载 `mytestdb.db` 数据库，但禁止任何用户使用该服务器来创建或删除数据库或者执行任何其它文件管理语句，而不管用户是否具有资源创建权限或者能否装载实用程序数据库并连接到该数据库。

```
dbsrv11 -n TestSrv -gu none c:\mytestdb.db
```

要只允许知道实用程序数据库口令的用户执行文件管理语句，通过运行以下命令启动服务器。

```
dbsrv11 -n TestSrv -su secret -gu utility_db
```

以下命令将 `Interactive SQL` 作为客户端应用程序启动，连接到名为 `TestSrv` 的服务器、装载实用程序数据库并连接到用户。

```
dbisql -c "UID=DBA;PWD=secret;DBN=utility_db;ENG=TestSrv"
```

在成功执行上述命令之后，用户连接到该实用程序数据库，并能执行文件管理语句。

消除数据库

消除数据库之后，所有表和数据都将从磁盘中删除，其中包括记录数据库变更的事务日志。所有数据库文件都是只读的，以防止数据库文件被意外修改或删除。缺省情况下，需要 DBA 权限才能消除数据库。通过使用数据库服务器 `-gu` 选项可以更改所需的权限。请参见“[-gu 服务器选项](#)”一节第 188 页。

在 Sybase Central 中，可以使用 [\[消除数据库向导\]](#) 消除数据库。

在 Interactive SQL 中，可以使用 `DROP DATABASE` 语句来消除数据库。

还可以使用 `dberase` 实用程序从命令行消除数据库。但是，`dberase` 实用程序不会消除 `dbspace`。如果要消除 `dbspace`，可以使用 `DROP DATABASE` 语句或 Sybase Central 中的 [\[消除数据库向导\]](#)。

使用 `dberase` 实用程序、[\[消除数据库向导\]](#) 或 `DROP DATABASE` 语句时，要消除的数据库不能处于运行状态。必须连接到某个数据库才能删除另一数据库。

有关连接到实用程序数据库的信息，请参见“[连接到实用程序数据库](#)”一节第 29 页。

Windows Mobile 数据库必须手工消除。请参见“[消除 Windows Mobile 数据库](#)”一节第 328 页。

◆ 消除数据库 (Sybase Central)

1. 选择 [\[工具\]](#) » [\[SQL Anywhere 11\]](#) » [\[消除数据库\]](#)。
2. 请按照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 [\[消除数据库向导\]](#)：

- 选择服务器服务器，然后选择 [\[文件\]](#) » [\[消除数据库\]](#)。
- 右击服务器，然后选择 [\[消除数据库\]](#)。

◆ 消除数据库 (SQL)

1. 连接到要消除的数据库以外的其它数据库。例如，连接到实用程序数据库。
2. 执行 `DROP DATABASE` 语句。

例如，下面的 `DROP DATABASE` 语句消除名为 `temp` 的数据库。

```
DROP DATABASE 'c:\\temp\\temp.db';
```

请参见“[DROP DATABASE 语句](#)”一节《[SQL Anywhere 服务器 - SQL 参考](#)》。

◆ 消除数据库 (命令行)

- 运行 `dberase` 实用程序。

例如，下面的命令删除 `temp` 数据库。

```
dberase c:\temp\temp.db
```

请参见“消除实用程序 (dberase)”一节第 722 页。

运行数据库服务器

目录

运行 SQL Anywhere 数据库服务器简介	36
启动数据库服务器	40
一些常见的选项	43
停止数据库服务器	53
启动和停止数据库	54
在当前会话外部运行服务器	57
对服务器启动进行故障排除	68
运行经过验证的 SQL Anywhere 应用程序	70
运行 SQL Anywhere Web 版应用程序	75
SQL Anywhere 中的错误报告	76

运行 SQL Anywhere 数据库服务器简介

SQL Anywhere 提供了两种版本的数据库服务器：

- **个人数据库服务器** 此可执行文件不支持通过网络进行客户端/服务器通信。尽管提供个人数据库服务器的目的是供单个用户在同一台计算机上使用—例如，作为嵌入式数据库服务器—但它对开发工作也有帮助。

在除 Windows Mobile 之外的 Windows 操作系统上，该个人服务器可执行文件的名称是 *dbeng11.exe*。在 Unix 操作系统上，其名称是 *dbeng11*。Windows Mobile 上仅支持网络服务器。

- **网络数据库服务器** 此可执行文件支持通过网络进行客户端/服务器通信，并可供多用户使用。

在包括 Windows Mobile 在内的 Windows 操作系统上，该网络服务器可执行文件的名称是 *dbsrv11.exe*。在 Linux 和 Unix 操作系统上，其名称为 *dbsrv11*。

服务器之间的差异

个人服务器和网络服务器上的请求处理引擎完全相同。每种服务器都支持完全相同的 SQL，并且具有完全相同的数据库功能。通过个人数据库服务器创建的数据库可在网络数据库服务器中使用，反之亦然。二者的主要区别包括：

- **网络协议支持** 只有网络服务器才支持通过网络进行通信。
- **连接数** 个人服务器的同时连接数上限为 10 个。网络服务器的限制取决于您的许可。请参见“[服务器授权实用程序 \(dblic\)](#)”一节第 766 页。
- **CPU 数** 对于每存取设备授权，网络数据库服务器使用计算机上可用的所有 CPU（缺省设置）。对于基于 CPU 的授权，网络数据库服务器仅使用您得到授权的处理器。网络数据库服务器可以使用的 CPU 数量还可能会受到 SQL Anywhere 版本或 `-gt` 服务器选项的影响。个人数据库服务器能够使用的处理器数量限制为一个。请参见：
 - “[版本和授权](#)”一节 《[SQL Anywhere 11 - 简介](#)》
 - “[-gt 服务器选项](#)”一节第 185 页
- **启动缺省值** 为了反映出它们是用作个人服务器还是用作供多名用户使用的网络服务器，彼此的启动缺省值稍有不同。

网络软件要求

如果要运行 SQL Anywhere 网络服务器，则必须安装相应的联网软件并运行该软件。

Windows、Linux 和 Unix 操作系统可以使用 SQL Anywhere 网络服务器。

SQL Anywhere 支持 TCP/IP 网络协议。

准备步骤

可通过以下几种方式来启动运行单个数据库的个人服务器：

- 在 Windows 上，从 [开始] 菜单中选择 [程序] » [SQL Anywhere 11] » [SQL Anywhere] » [个人服务器示例]。
- 在 *demo.db* 所在的目录中执行以下命令来启动个人服务器和名为 *demo.db* 的数据库：

```
dbeng11 demo
```

- 在连接字符串中使用数据库文件名。
请参见“[连接到嵌入式数据库](#)”一节第 118 页。

在何处指定命令

可以通过几种方式指定命令，具体视操作系统而定：

- 在命令提示符处运行命令。
- 将命令放在快捷方式或桌面图标中。
- 在批处理文件中运行命令。
- 将命令作为 StartLine (START) 连接参数包括在连接字符串内。请参见“[StartLine 连接参数 \[START\]](#)”一节第 281 页。

指定基本命令的方式因平台不同而略有差异。

启动数据库服务器

启动数据库服务器的方式因所使用的操作系统的不同而略有差异。本节介绍如何在所支持的每一种操作系统上针对以缺省设置运行单个数据库这种简单情况指定命令。

注意

- 除非另行说明，否则这些命令会启动个人服务器 (**dbeng11**)。要启动网络服务器，需将 **dbeng11** 替换为 **dbsrv11**。
- 如果数据库文件位于该命令的启动目录中，则无需指定 *path*。
- 如果未指定 *database-file* 的文件扩展名，将假定扩展名为 *.db*。

◆ 使用缺省选项启动个人数据库服务器（适用于除 Windows Mobile 以外的 Windows）

- 运行以下命令：

```
dbeng11 path\database-file
```

如果忽略数据库文件，将出现 [服务器启动选项] 窗口，可在其中单击 [浏览] 来查找数据库文件。

有关在 Windows Mobile 上启动数据库服务器的详细信息，请参见“[连接到在 Windows Mobile 设备上运行的数据库](#)”一节第 318 页。

◆ 使用缺省选项启动个人数据库服务器 (Unix)

- 运行以下命令：

```
dbeng11 path/database-file
```

还有哪些方法？

尽管可以按上一节介绍的简单方法启动个人服务器，但在生产环境中运行数据库服务器还有许多其它方面。例如，

- 有许多选项可供选择，可以通过这些选项指定多种功能，如用作高速缓存的内存量、要使用的 CPU 数量（在运行网络数据库服务器的多处理器计算机上）以及要使用的网络协议（仅限网络服务器）。这些选项是调整 SQL Anywhere 行为和性能的主要方法之一。请参见“[SQL Anywhere 数据库服务器](#)”一节第 148 页。
- 可以将服务器作为 Windows 服务来运行。如果将服务器作为服务运行，那么即使注销计算机，该服务器还将继续运行。请参见“[在当前会话外部运行服务器](#)”一节第 57 页。
- 可以从应用程序启动个人服务器并在应用程序使用完服务器之后将其关闭。这是在将数据库服务器用作嵌入式数据库时的通常做法。请参见“[连接到嵌入式数据库](#)”一节第 118 页。

在 Windows Vista 上运行 SQL Anywhere

SQL Anywhere 支持 Windows Vista 操作系统。以下是在 Vista 上运行 SQL Anywhere 软件的一些注意事项：

- **Vista 安全性** Vista 并入了称为用户帐户控制（User Account Control，简称 UAC）的新安全模型。UAC 将在缺省情况下启用，它可能会影响预期能够写入文件的程序的行为，尤其是在计算机支持多个用户的情况下。一位用户创建的某个文件可能具有禁止其他用户对该文件进行读写的权限，这取决于创建文件和目录的位置和方式。如果在缺省目录中安装了 SQL Anywhere，则会相应建立需要多个用户的读/写访问权限的文件和目录。
- **SQL Anywhere 已提升的操作代理** 在 Vista 中，某些操作需要提升权限才能在于 UAC 下运行时执行。以下程序在 SQL Anywhere 中可能需要提升：
 - *dbdsn.exe*
 - *dbelevat11.exe*
 - *dblic.exe*
 - *dbsvc.exe*
 - *installULNet.exe*
 - *mlasinst.exe*
 - *SetupVSPackage.exe*
 - *ulcond11.exe*

以下 DLL 在注册或注销时需要提升：

- *dbctrs11.dll*
- *dbodbc11.dll*
- *dboledb11.dll*
- *dboledba11.dll*

在已激活 UAC 的 Vista 系统上，可能会收到 SQL Anywhere 已提升操作代理的提升提示。此提示由 Vista 用户帐户控制系统发出，以确认要继续运行已标识的程序（如果以管理员身份登录）或提供管理员证书（如果以非管理员身份登录）。

- **部署注意事项** 程序 *dbelevate11.exe* 由 SQL Anywhere 组件内部使用，以执行需要提升权限的操作。这个可执行文件必须包含在 SQL Anywhere 部署中。
- **ActiveSync 支持** Vista 中不支持 Microsoft ActiveSync 实用程序。它由 Windows 移动设备中心所代替。您可以将 SQL Anywhere ActiveSync 提供程序安装实用程序与 Windows 移动设备中心结合使用。
- **SQL Anywhere 可执行文件已签名** Vista 上的 SQL Anywhere 可执行文件由 iAnywhere Solutions, Inc. 签名
- **Windows 服务** 符合 Vista 的服务不允许与桌面交互。在 Windows Vista 上，没有 SQL Anywhere 服务与桌面交互（即使服务定义中启用了 [允许服务与桌面交互] 也是如此）。可通过 Sybase Central 或 dbconsole 实用程序来监控 SQL Anywhere 数据库服务器。请参见“[SQL Anywhere 控制台实用程序 \(dbconsole\)](#)”一节第 780 页。

Sybase Central 禁用此选项，以使服务在 Windows Vista 上运行时能够与桌面交互。

- **使用 AWE 高速缓存** 要在 Vista 上使用 AWE 高速缓存，必须以管理员身份运行数据库服务器。使用 AWE 高速缓存启动未提升的数据库服务器，将会出现警告，提醒您必须以管理员身份运行数据库服务器才能使用 AWE。请参见“[-cw 服务器选项](#)”一节第 167 页。

启动数据库服务器

服务器命令的一般格式如下：

```
executable [ server-options ] [ database-file [ database-options ], ...]
```

如果不提供选项和数据库文件，则在 Windows 操作系统中会出现一个窗口，用于浏览查找您的数据库文件。

数据库服务器命令的元素包括：

- **可执行文件** 个人服务器 (dbeng11) 或网络服务器 (dbsrv11)。有关不同操作系统上的可执行文件名的详细信息，请参见“[运行 SQL Anywhere 数据库服务器简介](#)”一节第 36 页。
- **服务器选项** 这些选项为所有正在运行的数据库控制数据库服务器的行为。
- **数据库文件** 可以不指定数据库文件名，也可以指定一个或多个。这些数据库中的每一个都会启动，并保持可用状态，以供应用程序使用。

小心

数据库文件与事务日志文件必须与数据库服务器位于同一台物理计算机上，或者必须通过 SAN 或 iSCSI 配置进行访问。如果数据库文件和事务日志文件位于远程网络目录上，则可能会导致性能很差、数据损坏和服务器不稳定。

有关详细信息，请访问 <http://www.sybase.com/detail?id=1034790>。

为获得最佳结果，事务日志与数据库文件应保留在不同的磁盘上。请参见“[事务日志](#)”一节第 14 页。

- **数据库选项** 对于所启动的每个数据库文件，您可以提供用于控制其某些行为的数据库选项。请参见“[SQL Anywhere 数据库服务器](#)”一节第 148 页。

区分大小写

数据库和服务器选项通常区分大小写。应以小写形式输入所有选项。

列出可用的选项

◆ 列出数据库服务器的选项：

- 运行以下命令：

```
dbeng11 -?
```

记录数据库服务器操作

数据库服务器消息日志包含信息性消息、错误、警告以及来自 MESSAGE 语句的消息。记录服务器在开发过程中和进行故障排除时所执行的操作将会对您有所帮助。

这些消息可出现在以下位置：

- 数据库服务器消息窗口（在 Windows 中显示为系统任务栏图标）中
- Sybase Central 的 [服务器消息和已执行的 SQL] 窗格中
- SQL Anywhere 控制台实用程序中
- 数据库服务器消息日志文件中
- 在将数据库服务器作为命令行应用程序运行时，出现在命令提示窗口或 shell 中
- Unix Syslog 中

另请参见

- “-o 服务器选项”一节第 196 页
- “-oe 服务器选项”一节第 197 页
- “-on 服务器选项”一节第 198 页
- “-os 服务器选项”一节第 198 页
- “-ot 服务器选项”一节第 199 页

将数据库服务器消息记录到文件

缺省情况下，数据库服务器消息将被发送到数据库服务器消息窗口中。此外，也可以使用 `-o` 选项将输出发送到日志文件中。以下命令会将输出发送到名为 `mydbserver_messages.txt` 的日志文件中：

```
dbsrv11 -o mydbserver_messages.txt -c ...
```

可以控制数据库服务器消息日志文件的大小，并可指定在文件大小达到其最大值后要执行的操作：

- 使用 `-o` 选项指定应使用的数据库服务器消息日志文件并提供名称。
- 如果想先删除日志文件以前的内容，然后再向其发送消息，请使用 `-ot` 选项指定应使用的数据库服务器消息日志文件并提供名称。
- 除 `-o` 或 `-ot` 之外，还可以使用 `-on` 选项来指定一个大小，数据库服务器消息日志文件达到该大小时将以扩展名 `.old` 对其进行重命名，然后使用原始名称启动新文件。
- 除 `-o` 或 `-ot` 之外，还可以使用 `-os` 选项来指定一个大小，数据库服务器消息日志文件达到该大小时将启动新的日志文件，并根据日期和序号为新文件指定新名称。

可以使用 `-oe` 选项指定一个单独的文件来记录启动错误、致命错误和声明。

建议不要使用 `.log` 作为数据库服务器消息日志文件的扩展名，因为这将导致使用事务日志执行操作的实用程序发生问题。

另请参见

- “-o 服务器选项”一节第 196 页
- “-oe 服务器选项”一节第 197 页
- “-on 服务器选项”一节第 198 页
- “-os 服务器选项”一节第 198 页
- “-ot 服务器选项”一节第 199 页

在 Sybase Central 中记录 SQL 语句

当您在 Sybase Central 中使用数据库时，应用程序会根据您的操作自动生成 SQL 语句。您可以在一个名为 [服务器消息和已执行的 SQL] 的单独窗格中跟踪这些语句，或者将这些信息保存到文件中。每个数据库和数据库服务器在 [服务器消息和已执行的 SQL] 窗格中都有一个对应的选项卡。数据库服务器的选项卡与数据库服务器消息窗口包含相同的信息。

当您使用 Interactive SQL 时，还可以记录所执行的语句。请参见“记录命令”一节第 647 页。

◆ 记录 Sybase Central 生成的 SQL 语句

1. 选择 [视图] » [服务器消息和已执行的 SQL]。
2. 在 [服务器消息和已执行的 SQL] 窗格中，单击带数据库图标的选项卡。
3. 右击并选择 [选项]。
4. 编辑记录选项。
5. 单击 [保存]。
6. 选择保存文件的位置，然后单击 [确定]。
7. 单击 [确定]。

取消 Windows 事件日志消息

可通过设置一个注册表条目来取消 Windows 事件日志条目。该注册表条目是 *Software\Sybase\SQL Anywhere\11.0*。可将此条目置于 HKEY_CURRENT_USER 或 HKEY_LOCAL_MACHINE 配置单元中。

要控制事件日志条目，可设置 EventLogMask 项，它属于 REG_DWORD 类型。该值是一个位掩码，其中包含了不同类型事件消息的内部位值：

```
errors          EVENTLOG_ERROR_TYPE          0x0001
warnings       EVENTLOG_WARNING_TYPE         0x0002
information    EVENTLOG_INFORMATION_TYPE    0x0004
```

例如，如果将 EventLogMask 项设置为 0，则不显示任何消息。如果将该项设置为 1，则不显示信息性消息和警告消息，而只显示错误消息。缺省设置（无条目）可显示所有类型的消息。

更改 EventLogMask 项的设置后，只有重新启动数据库服务器才能使更改生效。

另请参见

- “网络协议选项”一节第 284 页

一些常见的选项

某些最常用的选项用于控制以下设置：

- 使用配置文件
- 命名服务器和数据库
- 性能
- 权限
- 最大页面大小
- 特殊模式
- 线程
- 网络通信（仅限网络服务器）

使用配置文件存储服务器启动选项

如果所使用的一组选项数量很多，可以将它们存储在配置文件中，然后在服务器命令中调用该文件。在配置文件中，选项可放在多行上。例如，下面的配置文件启动个人数据库服务器和示例数据库。它将高速缓存设置为 10 MB，并将此个人服务器实例命名为 **Elora**。以 # 字符开头的行将被视为注释。

```
# Configuration file for server Elora
-n Elora
-c 10M
samples-dir\demo.db
```

在本例中，*samples-dir* 是 SQL Anywhere 示例目录的名称。在 Unix 上，要在文件路径中使用正斜线而非反斜线。

有关 *samples-dir* 的信息，请参见“[示例目录](#)”一节第 368 页。

如果将该文件命名为 *sample.cfg*，则可以按以下所示使用这些选项：

```
dbeng11 @sample.cfg
```

另请参见

- “[@data 服务器选项](#)”一节第 156 页
- “[使用配置文件](#)”一节第 695 页
- “[在配置文件中使用条件分析](#)”一节第 696 页

命名服务器和数据库

可以将 `-n` 用作服务器选项（以对服务器命名）或数据库选项（以对数据库命名）。

客户端应用程序连接到数据库时可能会使用的连接参数中包括服务器名和数据库名。服务器名出现在桌面图标上和数据库服务器消息窗口的标题栏中。

命名服务器

提供数据库服务器名有助于避免与网络上的其它服务器名发生冲突。它也为客户端应用程序的用户提供了更有意义的名称。服务器在其生存期内保留其名称（直到它关闭）。如果不提供服务器名，将以所启动的第一个数据库的名称对服务器命名。

可以通过在第一个数据库文件之前提供一个 `-n` 选项来对服务器命名。例如，以下命令会启动示例数据库上的一个服务器并指定 `Cambridge` 作为它的名称：

```
dbeng11 -n Cambridge samples-dir\demo.db
```

如果提供服务器名，则可以在不启动数据库的情况下启动数据库服务器。以下命令会在不启动数据库的情况下启动名为 `Galt` 的服务器：

```
dbeng11 -n Galt
```

服务器名的最大长度为 250 个字节。

有关在运行中的服务器上启动数据库的详细信息，请参见“[启动和停止数据库](#)”一节第 54 页。

注意

对于 Windows 和 Unix，版本 9.0.2 和早期版本的客户端不能连接到名称超过以下长度的 10.0.0 版和更高版本的数据库服务器：

- 对于 Windows 共享内存，为 40 字节
- 对于 Unix 共享内存，为 31 字节
- 对于 TCP/IP，为 40 字节

命名数据库

您最好为客户端应用程序的用户提供一个有意义的数据库名。在停止数据库之前将一直使用该名称标识数据库。数据库名的最大长度为 250 个字节。

如果不提供数据库名，缺省名称将是数据库文件的根（即没有 `.db` 扩展名的文件名）。例如，在以下命令中，第一个数据库命名为 `mydata`，第二个数据库命名为 `mysales`。

```
dbeng11 c:\mydata.db c:\sales\mysales.db
```

可以通过在数据库文件后提供一个 `-n` 选项来对数据库命名。例如，以下命令会启动示例数据库并将其命名为 `MyDB`：

```
dbeng11 samples-dir\demo.db -n MyDB
```

区分大小写

只要字符集是单字节，服务器名和数据库名就不区分大小写。请参见“[连接字符串和字符集](#)”一节第 386 页。

从命令行控制性能和内存

有几个选项会对数据库服务器性能产生重要的影响，其中包括：

- **高速缓存大小** 数据库服务器可用的高速缓存量可能是影响性能的关键因素。数据库服务器采用的高速缓存的初始值，可以由 `-c` 选项指定，也可以采用缺省值。

`-c` 选项控制 SQL Anywhere 用作高速缓存的内存量。

一般来说，数据库服务器可以使用的内存越多，其执行的速度就越快。高速缓存用于保存可能需要多次使用的信息。访问高速缓存中的信息比从磁盘中访问同样的信息快数倍。缺省的初始高速缓存大小是基于物理内存量、操作系统和数据库文件的大小计算出的。在 Windows 和 Unix 操作系统上，可用的高速缓存耗尽时数据库服务器会自动增加高速缓存。

数据库服务器消息窗口显示启动时的高速缓存大小，可以使用以下语句来获得当前高速缓存的大小：

```
SELECT PROPERTY( 'CacheSize' );
```

有关性能调优的详细信息，请参见“[提高数据库性能](#)”《SQL Anywhere 服务器 - SQL 的用法》。

有关控制高速缓存大小的详细信息，请参见“[-c 服务器选项](#)”一节第 158 页。

在 Windows 和 Unix 上，数据库服务器会根据需要（由启发式算法确定）自动将更多内存用于高速缓存。请参见“[使用高速缓存提高性能](#)”一节《SQL Anywhere 服务器 - SQL 的用法》。

您可以使用数据库选项来配置高速缓存上限。请参见“[-ch 服务器选项](#)”一节第 161 页。

此外，还可以强制高速缓存保持其初始值。请参见“[-ca 服务器选项](#)”一节第 159 页。

- **进程并发水平** 数据库服务器的进程并发水平是指可并发执行的最大服务器任务数。一般而言，如果进程并发水平较高，则由于允许更多的请求同时执行，因此会提高服务器的总吞吐量。不过，如果请求为获得相同的资源而竞争，则提高进程并发水平可能会导致更多争用，实际上会延长事务响应时间。

在某些情况下，提高进程并发水平甚至会降低系统的吞吐量。可以使用 `-gn` 选项来设置服务器的进程并发水平。请参见“[-gn 服务器选项](#)”一节第 182 页和“[设置数据库服务器的进程并发水平](#)”一节第 50 页。

- **处理器数** 如果运行的是使用网络数据库服务器的多处理器计算机，则可以使用 `-gt` 选项来设置处理器的数量。请参见“[-gt 服务器选项](#)”一节第 185 页和“[SQL Anywhere 中的线程](#)”一节第 47 页。

数据库服务器可以使用的 CPU 数量还可能会受到您的许可或 SQL Anywhere 版本的影响。请参见“[版本和授权](#)”一节《SQL Anywhere 11 - 简介》。

- **其它与性能有关的选项** 可使用多个选项来调整网络性能，包括 `-gb`（数据库处理优先级）和 `-u`（缓冲磁盘 I/O）。请参见“[SQL Anywhere 数据库服务器](#)”一节第 148 页。

从命令行控制权限

一些选项控制执行某些全局操作所需的权限，其中包括启动和停止数据库、装载和卸载数据以及创建和删除数据库文件的权限。请参见“[以安全的方式运行数据库服务器](#)”一节第 1018 页。

设置最大页面大小

数据库服务器高速缓存按页（内存中大小固定的区域）进行组织。由于服务器在其生存期内（直到它被关闭）只使用一个高速缓存，因此所有页的大小必须相同。

数据库文件也是按页进行组织，页的大小在命令行中指定。每个数据库页都必须与高速缓存页相匹配。在缺省情况下，服务器的页面大小与命令行上各个数据库的最大页面大小相等。服务器启动之后，即无法启动页面大小超过服务器页面大小的数据库。

要在服务器启动后还能够启动页面大小超过服务器页面大小的数据库，可以使用 `-gp` 选项来强制服务器以指定的页面大小启动。如果您使用较大的页面大小，切记增加您的高速缓存大小。由于相同大小的高速缓存只能容纳少许较大页面，因而降低了空间安排的灵活性。

以下命令会启动一个保留 64 MB 高速缓存的服务器，该高速缓存最多可以容纳页面大小为 8192 字节的数据。

```
dbsrv11 -gp 8192 -c 64M -n myserver
```

在特殊模式下运行

可以针对特定目的在特殊模式下运行 SQL Anywhere。

- **只读** 可以提供 `-r` 选项，以在只读模式下运行数据库。不能以只读模式启动开启了审计的数据库。请参见“[-r 服务器选项](#)”一节第 204 页和“[-r 数据库选项](#)”一节第 240 页。
- **内存模式** 通过指定 `-im` 选项，可以完全地在内存中运行数据库。当仅以检查点模式 (`-im c`) 运行时，数据库服务器不使用事务日志，但可将数据库恢复到最近的检查点。当以从不写入模式 (`-im nw`) 运行数据库时，已提交事务将不会写入磁盘上的数据库文件，并且在关闭数据库时所有更改都将丢失。而使用内存模式时，应用程序仍可更改数据库，或在数据库处于活动状态时对其进行访问。请参见“[-im 服务器选项](#)”一节第 188 页。

需要单独授予许可的组成部分

内存模式需要一个单独的许可。请参见“[单独授权的组件](#)”一节《SQL Anywhere 11 - 简介》。

- **批量装载** 在使用 Interactive SQL INPUT 命令将大量数据装载到数据库中时，此选项会很有用。如果使用 LOAD TABLE 来批量装载数据，请不要使用 `-b` 选项。请参见“[-b 服务器选项](#)”一节第 157 页和“[导入和导出数据](#)”《SQL Anywhere 服务器 - SQL 的用法》。
- **在没有事务日志的情况下启动** 使用 `-f` 数据库选项进行恢复—或者强制数据库服务器在事务日志丢失后启动，或者强制数据库服务器使用在其它情况下无法找到的事务日志来启动。请注意，`-f` 是数据库选项，而非服务器选项。

在恢复完成后，您应当停止服务器并在没有 `-f` 选项的情况下重新启动服务器。请参见“[-f 恢复选项](#)”一节第 175 页。

- **安静地运行** 数据库服务器支持安静模式。由您来决定服务器运行时的安静程度，从取消消息或系统任务栏中的图标直至完全安静。要在 Windows 上完全安静地运行数据库服务器，请指定 `-qi`、`-qs` 和 `-qw` 选项。设置了这些选项后，因为取消了所有图标和所有可能出现的启动错误消息，将看不到服务器正在运行的任何指示。在安静模式下运行数据库服务器时，可以使用 `-o` 和/或 `-oe` 选项来诊断错误。

注意，`-qi` 和 `-qs` 选项不会取消因使用 `-v`（版本）和 `-ep`（数据库加密口令提示）服务器选项而出现的窗口。

SQL Anywhere 中的线程

要理解 SQL Anywhere 线程模型，还必须了解线程和请求处理的基本术语和概念：

- **请求** 请求是一种通过连接发送到数据库服务器的工作单元，如查询或 SQL 语句。请求的生存期始于数据库服务器首次接收请求时，终于返回最后一条结果、关闭游标或取消请求时。
- **任务** 任务是一种在数据库服务器内执行的活动单元，是服务器调度的最小工作单元。在数据库服务器内，每个用户请求至少会成为一项任务，如果涉及查询内并行机制，则可能会成为更多项任务。除用户请求外，数据库服务器还可以调度其自己的任务来执行日常内部整理工作，如运行清理程序或处理计时器。可并发执行的最大活动任务数由 `-gn` 选项来设置。如果到达数据库服务器的任务数量超过了可以并发处理的任务数量，则会将多出的任务排入队列中，等待执行。如果由于某种原因（如等待锁或等待 I/O 完成）需要在对活动任务或已开始处理的任务进行处理的过程中将其阻塞，则该任务仍将被视为活动任务。因此，它会减少 `-gn` 选项值所指定的上限。
- **线程** 线程是一种表示应用程序内正在执行中的一连串控制的操作系统构造。每个操作系统进程（包括数据库服务器）都至少由一个线程来执行，可能的情况下会由许多线程来执行。操作系统在应用程序外调度线程，应用程序的所有执行操作最终都由其线程来执行。SQL Anywhere 数据库服务器内的任务在操作系统线程中执行。启动时，SQL Anywhere 会创建固定数量的线程，这些线程由 `-gtc` 选项控制（在 Windows 和 Linux 上），或由 `-gn` 选项控制（在 Unix 上）。

另请参见

- “控制线程行为”一节第 48 页
- “查询执行期间的并行”一节 《SQL Anywhere 服务器 - SQL 的用法》
- “`-gn` 服务器选项”一节第 182 页
- “`-gtc` 服务器选项”一节第 186 页
- “`sa_clean_database` 系统过程”一节 《SQL Anywhere 服务器 - SQL 参考》
- “事务阻塞和死锁”一节 《SQL Anywhere 服务器 - SQL 的用法》

Unix 上的任务

在 Unix 上，任务直接在操作系统线程中执行。在这些平台上，`-gn` 选项的值设置数据库服务器启动时所创建的操作系统线程数量；此组线程将为所有任务提供服务。线程变为可用状态时，就会拾取下一个需要处理的可用任务。处理任务时，线程将一直用于该任务，直到任务处理完毕。如果由于某种原因需要阻塞任务（可能是由于该任务正在等待 I/O 操作或等待锁），则线程会主动放弃对 CPU 的控制，将控制权交还操作系统调度程序，以使其它线程能够在该 CPU 上运行。

除主动放弃对 CPU 的控制外，线程还可能会被操作系统调度程序搁置。进程内的每个应用程序线程都分配有一系列运行时间片，其长度由其优先级和其它系统因素决定。线程运行到其当前时间片结束时，操作系统会搁置线程并安排其过一段时间再次运行。操作系统调度程序接着选择另一个线

程来执行一个时间片。这种抢先调度不会对任务处理产生任何可见的影响；线程被安排再次运行时，将从停止处理的位置继续对任务进行处理。

对活动任务的处理完成后，线程即会进行检查以确认是否有任何其它任务可供处理。如果有，线程会拾取下一个可用任务，然后继续。否则，它会放弃对 CPU 的控制，等待新任务到达数据库服务器。

另请参见

- “-gn 服务器选项”一节第 182 页

Windows 和 Linux 上的任务

在 Windows 和 Linux 上，任务在称为**纤程**的轻量线程中执行。纤程使线程上运行的任务能够以协作方式自行调度，而不依赖操作系统的线程调度程序。结果是，纤程间的上下文切换开销远低于线程上下文切换开销，因为不需要与操作系统内核或调度程序进行交互。不使用纤程时，多线程应用程序会频繁进行线程上下文切换，而使用纤程后，则可以显著提高性能和可伸缩性。

由于纤程不依赖操作系统调度程序，因此在其等待某个其它活动完成时，必须显式地将控制权让与另一个纤程。例如，如果等待 I/O 操作完成时需要阻塞某个正在纤程中执行的任务，则当前纤程会将控制权让与另一个纤程。托管原始纤程的线程可以立即拾取另一个纤程并开始执行，而不需要进行内核上下文切换。如果某个纤程产生阻塞但不让出控制，则它会阻塞托管它的线程，并禁止其它纤程在该线程上运行。如果有多个线程在托管纤程，则只有托管等待中纤程的线程会被阻塞：其它线程仍可自由地运行纤程。

在支持纤程的平台上，所创建的纤程数量至少要与服务器最大并发数设置所要求的数量（即 -gn 选项所指定的数量）相同。服务器创建的纤程数量可以多于此值，这样一来，始终都会有纤程可以为服务器内部任务提供服务。请参见“-gn 服务器选项”一节第 182 页。

控制线程行为

控制线程行为的主要因素有五个，每个因素都由一个服务器选项来控制。并非每个平台都支持所有这些选项。

- **进程并发水平（-gn 服务器选项）** -gn 选项控制服务器的进程并发水平。此值决定可以同时处于活动状态的最大任务数量。每个数据库请求至少会使用一项任务，如果涉及查询内并行机制，则可能会使用更多项任务。此外，服务器有时会安排任务去执行内部整理活动。服务器中的任务数量超过进程并发水平时，待处理的任务必须等待，直到当前运行的任务或活动任务完成。缺省情况下，网络数据库服务器和个人数据库服务器最多可以并发执行 20 项任务。请参见“-gn 服务器选项”一节第 182 页和“设置数据库服务器的进程并发水平”一节第 50 页。
- **每个内部执行线程的堆栈大小（-gss 服务器选项）** 可以使用 -gss 选项设置服务器中每个内部执行线程的堆栈大小。使用 -gss 选项可以降低数据库服务器的内存使用量，在内存有限的环境中该选项可能会有帮助。Windows Mobile 是唯一支持此选项的 Windows 操作系统。请参见“-gss 服务器选项”一节第 185 页。
- **处理器数量（-gt 服务器选项）** 如果有多个处理器，则可以通过指定 -gt 选项来控制线程利用的处理器数量。请参见“-gt 服务器选项”一节第 185 页。

- **处理器并发数 (-gtc 服务器选项)** 可以指定最多能够在 CPU 上并发运行的线程数量。缺省情况下，数据库服务器会在每个授权的物理处理器的所有超线程和内核上运行。请参见“[-gtc 服务器选项](#)”一节第 186 页。

线程技巧

- 增大 -gn 可以降低发生线程死锁的几率。请参见“[-gn 服务器选项](#)”一节第 182 页。
- 将 -gt 设置为 1 有助于以变通方式解决并发性问题。请参见“[-gt 服务器选项](#)”一节第 185 页。
- 调查性能监控器中 [请求: 活动] 和 [请求: 未调度] 的读数有助于确定适合于 Windows 的 -gn 值。如果活动请求的数量始终小于 -gn，则可以降低 -gn。如果请求总数（活动请求数与未调度请求数之和）经常大于 -gn，则可能需要增加 -gn 的值。请参见“[性能监控器统计](#)”一节《[SQL Anywhere 服务器 - SQL 的用法](#)》和“[-gn 服务器选项](#)”一节第 182 页。

处理器使用和线程示例

以下示例说明数据库服务器如何根据 -gt 和 -gtc 的设置来选择 CPU 数量。为便于说明以下示例，假定系统安装有 4 个处理器，每个处理器有 2 个内核。物理处理器以字母进行标识，内核以数字进行标识，于是此系统具有处理单元 A0、A1、B0、B1、C0、C1、D0 和 D1。

方案	网络数据库服务器设置
单 CPU 许可或将 -gt 指定为 1	<ul style="list-style-type: none"> ● -gt 1 ● -gtc 2 ● -gn 20 <p>线程可以在 A0 和 A1 上执行。</p>
无 CPU 许可数量限制，并将 -gtc 指定为 5	<ul style="list-style-type: none"> ● -gt 4 ● -gtc 5 ● -gn 20 <p>线程可以在 A0、A1、B0、C0 和 D0 上执行。</p>
具有 3 CPU 许可并指定了 -gtc 5 的数据库服务器	<ul style="list-style-type: none"> ● -gt 3 ● -gtc 5 ● -gn 20 <p>线程可以在 A0、A1、B0、B1 和 C0 上执行。</p>
无 CPU 许可数量限制，并将 -gtc 指定为 1	<ul style="list-style-type: none"> ● -gt 4 ● -gtc 1 ● -gn 20 <p>线程只能在 A0 上执行。</p>

设置数据库服务器的进程并发水平

数据库服务器的**进程并发水平**是指最多可以同时处于活动状态的任务数量，通过 `-gn` 服务器选项进行控制。活动任务是指数据库服务器中当前正由线程（或纤程）执行的任务。活动任务可能正在执行访问计划运算符，或执行某个其它有用的工作，但也可能因等待资源（如 I/O 操作或某行上的锁）而被阻塞。未调度任务是指可以执行但正在等待可用线程或纤程的任务。可以同时执行的活动任务数量取决于计算机上使用中的数据库服务器线程数量和逻辑处理器数量。

服务器执行期间进程并发水平保持不变，并应用于该服务器上的所有数据库。对于网络数据库服务器和个人数据库服务器，其缺省值均为 20 个活动任务，但在 Windows Mobile 上例外，在该平台上其缺省值为 3 个活动任务。

提高进程并发水平

确定何时提高或降低进程并发水平可能是件困难的工作。例如，如果数据库应用程序利用 Java 存储过程或启用了查询内并行机制，则为处理这些请求而额外创建的服务器任务可能会超过进程并发限制，要等到另一请求完成后再执行这些任务。在这种情况下，提高进程并发水平可能是适宜的做法。通常，提高进程并发水平会相应地提高数据库服务器的总吞吐量，因为这样做将允许并发执行更多任务（请求）。不过，应考虑到提高进程并发水平会产生负面影响。这些负面影响包括：

- **增加了争用** 提高并发任务数量可能使活动请求之间发生争用的可能性增大。争用可能涉及模式或行锁等资源，或数据库服务器内部的数据结构和/或同步基元。此类状况实际上可能会降低服务器吞吐量。
- **增加了服务器开销** 每项活动任务都要求对线程（在 Windows 和 Linux 上是一种叫作纤程的轻量线程）进行分配和维护，并要求使用附加簿记结构来控制对它的调度。此外，每项活动任务都要求为其执行堆栈预分配地址空间。堆栈的大小因平台而异，但一般来说 32 位平台上为 1 MB，64 位平台上会更大。在 Windows 系统上，分配堆栈空间会影响服务器进程的地址空间，但堆栈内存是按需分配的。在 Unix 平台（包括 Linux）上，堆栈后备内存的分配是即时进行的。因此，设置较高的进程并发水平会增加服务器的内存占用并减少可以用作高速缓存的内存量，因为可用地址空间减少了。
- **系统失效** 数据库服务器可能会达到这样一种状态：将大量资源全部用于管理其执行开销，而不是为特定请求执行有用的工作。这种状态通常称为系统失效。举例来说，在以下情况下可能会出现系统失效的状况：为获取数据库高速缓存空间而竞争的活动请求数量过多，但高速缓存不足以容纳这组活动请求所使用的有效数据库页集。这种情况可能会导致页面挪用，其发生方式与操作系统中页面挪用的发生方式类似。
- **对查询处理有影响** 数据库服务器将选择可并发处理的最大数量的内存密集型请求。即便提高数据库服务器的进程并发水平，请求可能仍需要在内存可用之前一直等待。请参见“[内存调控器](#)”一节《[SQL Anywhere 服务器 - SQL 的用法](#)》。
- **数据结构占用内存** 数据库服务器需要使用资源来分析和优化语句。如果语句非常复杂或高速缓存过小，服务器数据结构所占用的内存可能超过可用内存量。内存调控器限制了每个任务的服务器数据结构所使用的内存量。每个任务都具有以下限制：

$$(3/4 \text{ maximum cache size}) / (\text{number of currently active tasks})$$

如果超过此限制，语句将失败并出现错误。

降低进程并发水平

通过减少并发执行的任务数量来降低数据库服务器的进程并发水平通常会减少服务器的吞吐量。不过，降低进程并发水平可能会缩短单个请求的响应时间，原因是为获得资源而竞争的请求数量减少了，发生锁争用的可能性也降低了。

在 SQL Anywhere 中，线程（纤程）以协作方式执行任务。完成任务后，线程（纤程）即可不受限制地拾取下一个等待执行的任务。不过，如果任务被阻塞（例如，等待行锁时），则线程（纤程）也会被阻塞。

当进程并发水平设置过低时，可能会发生**线程死锁**。假定数据库服务器有 n 个线程（纤程）。当 $n-1$ 个线程被阻塞，并且最后一个线程也即将被阻塞时，就会发生线程死锁。数据库服务器的内核不会允许这最后一个线程被阻塞，因为这样会导致所有线程都被阻塞，服务器将因此而挂起。数据库服务器会执行 SQLSTATE 40W06 来结束即将阻塞最后一个线程的任务。

如果就负载而言进程并发水平处于一个合理的级别，那么发生线程死锁就是应用程序存在设计问题的征兆，这种问题会导致大量争用，并因此削弱可伸缩性。例如，每个应用程序在向数据库插入新数据时都必须修改的表就是一种典型情况。此技术常用作模式的一部分来生成主键。但结果是，它对应用程序的所有插入事务进行了有效序列化。当由于共享表上的序列化而使插入事务的速率超过服务器能够提供服务的能力时，通常会发生线程死锁。

选择进程并发水平

建议对应用程序的负载进行试验，以分析服务器的进程并发水平对服务器吞吐量和请求响应时间的影响。有各种性能计数器可供使用，它们要么以属性函数形式出现，要么通过 Windows 上的 Windows 性能监控器提供，这些性能计数器可以在测试应用程序时帮助您分析数据库服务器的行为。对这种分析而言，与活动请求和未调度请求有关的性能计数器十分重要。

如果活动请求的数量总是低于 `-gn` 数据库服务器选项的值，则可以考虑降低进程并发水平，但必须将查询内并行机制的影响考虑在内，该机制会在服务器的执行队列中添加额外任务。如果查询内并行机制的影响很小，就可以放心地降低进程并发水平而不会降低系统的总吞吐量。不过，如果总请求数量（活动请求与未调度请求数量之和）经常大于 `-gn`，则提高进程并发水平可能是有充分根据的，但会出现前文中概要介绍的负面影响。请注意，Unix 或 Linux 平台上未提供性能监控器。

选择通信协议

客户端应用程序与数据库服务器之间的任何通信都需要使用通信协议。SQL Anywhere 支持一组通信协议，这些协议用于在网络之间进行通信以及在同一台计算机上进行通信。

缺省情况下，数据库服务器会启动所有可用的协议。可以使用 `-x` 选项限制数据库服务器可以使用的协议。在客户端，许多同样的选项可以使用 `CommLinks (LINKS)` 连接参数来控制。

有关使用这些选项运行服务器的详细信息，请参见“[支持的网络协议](#)”一节第 134 页。

个人服务器可以使用的协议

个人数据库服务器 (`dbeng11.exe`) 支持以下协议：

- **共享内存** 此协议用于在同一台计算机上进行通信，始终保持可用状态。大部分平台都提供了该协议，请参见 <http://www.sybase.com/detail?id=1062623>。

对于在相同计算机上的通信，[共享内存] 往往比 TCP/IP 提供的性能更好。

- **TCP/IP** 该协议用于同一台计算机上从 TDS 客户端、Open Client 或 jConnect JDBC 驱动程序进行的通信。如果想从 Open Client 或 jConnect 进行连接，则不能禁用 TCP/IP。

有关 TDS 客户端的详细信息，请参见“[将 SQL Anywhere 用作 Open Server](#)”第 1057 页。

网络服务器可以使用的协议

网络数据库服务器 (*dsrv11.exe*) 支持以下协议：

- **共享内存** 此协议用于在同一台计算机上进行通信，始终保持可用状态。所有平台都提供了该协议。
- **TCP/IP** 大部分平台都支持该协议，请参见 <http://www.sybase.com/detail?id=1062623>。

共享内存和终端服务

使用终端服务时，共享内存客户端只能找到在同一终端中运行的数据库服务器。如果将终端服务用于作为服务运行的数据库服务器，则只有在控制台上运行的客户端可以进行连接。在非控制台终端上运行的客户端无法通过共享内存进行连接。在上述情况下，可以不使用共享内存，而是使用 TCP/IP 以使客户端能够进行连接。

有关保护 Unix 上共享内存连接的信息，请参见“[安全提示](#)”一节第 1005 页。

指定协议

通过使用 -x 选项，可以指示数据库服务器只使用某些可用的网络协议。以下命令使用 TCP/IP 协议启动示例数据库：

```
dsrv11 -x "tcpip" samples-dir\demo.db
```

尽管本示例中未做严格要求，但如果 -x 的任何参数中有空格，则必须使用引号。

可以添加其它参数来针对每种协议调整服务器的行为。例如，以下命令（全部在一行中键入）指示服务器使用两块网卡，其中一块使用指定的端口号。

```
dsrv11 -x "tcpip(MyIP=192.75.209.12:2367,192.75.209.32)" samples-dir\demo.db
```

有关可与 -x 选项一起使用的可用网络协议选项的详细信息，请参见“[网络协议选项](#)”一节第 284 页。

停止数据库服务器

您可以通过下列方法停止数据库服务器：

- 在数据库服务器消息窗口中单击 **[关闭]**。
- 使用 `dbstop` 实用程序。

`dbstop` 实用程序在批处理文件中或停止另一台计算机上的服务器时十分有用。它在其命令中需要一个连接字符串。请参见“[停止服务器实用程序 \(dbstop\)](#)”一节第 783 页。

- 让服务器按缺省设置在应用程序断开连接时自动关闭。
- 在 Unix 系统上，当焦点位于数据库服务器消息窗口中时按 Q 键。

示例

◆ 使用 `dbstop` 实用程序停止服务器：

1. 启动服务器。例如，从 SQL Anywhere 安装目录执行的以下命令使用示例数据库启动一个名为 Ottawa 的服务器：

```
dbsrv11 -n Ottawa samples-dir\demo.db
```

2. 使用 `dbstop` 停止服务器：

```
dbstop -c "ENG=Ottawa;UID=DBA;PWD=sql"
```

谁可以停止服务器？

启动服务器时，可以使用 `-gk` 选项设置用户通过 `dbstop` 停止服务器时所需的权限级别。对于个人数据库服务器，缺省值为 `all`。对于网络数据库服务器，所需的缺省权限级别为 `DBA`，但也可以将该值设置为 `all` 或 `none`。（不过，任何可以操作该计算机的人员都可以在数据库服务器消息窗口中单击 **[关闭]**。）

关闭操作系统会话

如果在数据库服务器运行时关闭其所在的操作系统会话，或使用操作系统命令来停止数据库服务器，则服务器会关闭，但并未完全关闭。下次装载数据库时必须进行恢复，而且恢复会自动进行。

有关恢复的详细信息，请参见“[备份和数据恢复](#)”第 817 页。

较好的做法是，先显式地停止数据库服务器，然后再关闭操作系统会话。

不会将服务器完全停止的命令示例包括：

- 在 Windows 任务管理器中停止进程
- 使用 Unix 的 `slay` 或 `kill` 命令

启动和停止数据库

数据库服务器中可以同时装载多个数据库。可以按以下所示同时启动数据库和数据库服务器：

```
dbeng11 demo sample
```

小心

数据库文件必须位于数据库服务器所在的计算机上。管理位于网络驱动器上的数据库文件可能会导致文件损坏。

在正在运行的服务器上启动数据库

您还可以按照下列方式之一在启动服务器之后启动数据库：

- 在已连接到服务器的情况下使用 DatabaseFile (DBF) 连接参数连接到数据库。DatabaseFile (DBF) 连接参数为新连接指定数据库文件。数据库文件将在当前服务器上启动。
请参见“[连接到嵌入式数据库](#)”一节第 118 页或“[DatabaseFile 连接参数 \[DBF\]](#)”一节第 258 页。
- 在选择了服务器的情况下，使用 START DATABASE 语句或者从 Sybase Central 的 [文件] 菜单中选择 [启动数据库]。
请参见“[START DATABASE 语句](#)”一节《[SQL Anywhere 服务器 - SQL 参考](#)》。

限制

- 服务器使用固定大小的页将数据库信息保存在内存中。一旦服务器已经启动，您就不能启动其页面大小比服务器页大的数据库了。
请参见“[设置最大页面大小](#)”一节第 46 页。
- -gd 服务器选项决定了启动数据库所需的权限。

启动数据库

使用 Sybase Central 和 Interactive SQL，都可以在不连接数据库的情况下将其启动。

◆ 在不进行连接的情况下启动服务器上的数据库 (Sybase Central)

1. 选择服务器，然后选择 [文件] » [启动数据库]。
2. 在 [启动数据库] 窗口中，输入所需的值。
该数据库作为未连接的数据库出现在数据库服务器下面。

◆ 在不进行连接的情况下启动服务器上的数据库 (SQL)

- 执行 START DATABASE 语句。

请参见“[START DATABASE 语句](#)”一节《[SQL Anywhere 服务器 - SQL 参考](#)》。

在名为 `sample` 的数据库服务器上，启动数据库文件 `c:\temp\temp.db`。

```
START DATABASE 'c:\\temp\\temp.db'  
AS tempdb ON 'sample'  
AUTOSTOP OFF;
```

只有连接到某个数据库才能启动另一数据库。

AUTOSTOP OFF 会阻止数据库在所有连接都断开时自动停止。此处用它来说明稍后要讨论的一个要点。

有关启动数据库的详细信息，请参见“[运行数据库服务器](#)”第 35 页。

停止数据库

您可以通过如下方法停止数据库：

- 与由连接字符串启动的数据库断开连接。除非将 `AutoStop (ASTOP)` 连接参数显式地设置为 `NO`，否则会自动执行该操作。
请参见“[AutoStop 连接参数 \[ASTOP\]](#)”一节第 251 页。
- 使用 Interactive SQL 或嵌入式 SQL 中的 `STOP DATABASE` 语句。
请参见“[STOP DATABASE 语句](#)”一节《[SQL Anywhere 服务器 - SQL 参考](#)》。

使用 Sybase Central 和 Interactive SQL 都可以停止数据库在数据库服务器上的运行。但不能停止当前所连接的数据库。必须先断开与数据库的连接，然后再将数据库停止。只有先连接到同一数据库服务器上的另一数据库，然后才能停止当前数据库。

有关停止数据库的详细信息，请参见“[运行数据库服务器](#)”第 35 页。

◆ 断开连接后停止服务器上的数据库 (Sybase Central)

1. 确保已连接到同一数据库服务器上的至少一个其它数据库。如果服务器上没有运行其它数据库，可以连接到实用程序数据库。
2. 选择要停止的数据库，然后选择 [文件] » [停止数据库]。

断开数据库的连接后，数据库可能会从左窗格中消失。如果该连接是仅剩的唯一连接，并在启动数据库时指定了 `AUTOSTOP`，就会发生这种情况。`AUTOSTOP` 会使数据库在最后一个连接断开后自动停止。

◆ 断开连接后停止服务器上的数据库 (SQL)

1. 如果尚未连接到服务器上的任何数据库，则连接到某个数据库（例如实用程序数据库）。
2. 执行 `STOP DATABASE` 语句。

示例

以下语句连接到实用程序数据库并停止 `tempdb` 数据库。

```
CONNECT to 'TestEng' DATABASE utility_db  
AS conn2  
USER 'DBA'  
IDENTIFIED BY 'sql';  
STOP DATABASE tempdb;
```

只有连接到某个数据库才能停止另一数据库。

另请参见

- [“连接到实用程序数据库”一节第 29 页](#)

在当前会话外部运行服务器

使用用户 ID 和口令登录计算机时，就建立了一个**会话**。当您启动数据库服务器或任何其它应用程序时，它就在该会话内运行。注销计算机时，与该会话关联的所有应用程序都会关闭。

通常要求数据库服务器随时可用。可以运行 SQL Anywhere for Windows 和 SQL Anywhere for Unix，以便注销计算机后数据库服务器仍可运行。

- **Windows 服务** 可以将 Windows 数据库服务器作为服务来运行。这样配置对于运行高可用性服务器而言十分便利。请参见“[了解 Windows 服务](#)”一节第 59 页。
- **Unix 守护程序** 可以使用 `-ud` 选项将 Unix 数据库服务器作为守护程序来运行，从而使数据库服务器能够在后台运行，并能在用户注销之后继续运行。请参见“[将 Unix 数据库服务器作为守护程序来运行](#)”一节第 57 页。
- **Linux 服务** 可以将 Linux 数据库服务器作为服务来运行。这样配置对于运行高可用性服务器而言具有许多便利的特性。请参见“[用于 Linux 的服务实用程序 \(dbsvc\)](#)”一节第 769 页。

除了为 SQL Anywhere 数据库服务器创建服务外，还可为以下可执行文件创建 Windows 服务：

- SQL Anywhere 日志传送管理器 (LTM)
- SQL Remote 消息代理 (dbremote)
- MobiLink 服务器 (mlsrv11)
- MobiLink 同步客户端 (dbmlsync)
- rshost 实用程序 (rshost)
- RSOE
- SQL Anywhere 广播转发器 (dbns11)
- 监听器实用程序 (dblsn)

另请参见

- “[用于 Windows 的服务实用程序 \(dbsvc\)](#)”一节第 773 页

将 Unix 数据库服务器作为守护程序来运行

要在后台运行 Unix 数据库服务器，并使其能够独立于当前会话运行，可以将其作为**守护程序**来运行。

不要使用 '&' 在后台运行数据库服务器

如果使用 Unix `&`（和符号）命令在后台运行数据库服务器，将不会奏效—服务器将立即关闭或停止响应。必须改为将数据库服务器作为守护程序来运行。

同样，如果尝试在程序内使用典型的 `fork() -exec()` 序列在后台启动服务器，也不会奏效。如果需要这样做，请将 `-ud` 选项加入数据库服务器选项列表。

可以按以下方式之一将 Unix 数据库服务器作为守护程序来运行：

1. 启动数据库服务器时使用 `-ud` 选项。例如：

```
dbsrv11 -ud demo
```

2. 使用 `dbspawn` 工具来启动数据库服务器。例如：

```
dbspawn dbsrv11 demo
```

使用 `dbspawn` 的一个优点是，`dbspawn` 进程在确认守护程序已经启动并已准备好接受请求之后才会关闭。如果由于任何原因守护程序无法启动，`dbspawn` 的退出代码将不为零。

使用 `-ud` 选项直接启动守护程序时，在守护程序将自身初始化或尝试打开命令中指定的任一数据库之前，`dbeng11` 和 `dbsrv11` 命令会创建守护程序进程并立即返回（退出并允许下一个命令得到执行）。

如果想要确保护程序运行的一个或多个应用程序将会使用数据库服务器，可以使用 `dbspawn` 来确保护程序在启动应用程序之前就已运行。以下示例说明如何使用 `csh` 脚本对此进行测试。

```
#!/bin/csh
# start the server as a daemon and ensure that it is
# running before you start any applications
dbspawn dbsrv11 demo
if ( $status != 0 ) then
    echo Failed to start demo server
    exit
endif
# ok, now you can start the applications
...
```

本示例使用 `sh` 脚本来测试守护程序是否在启动应用程序之前就已运行。

```
#!/bin/sh
# start the server as a daemon and ensure that it is
# running before you start any applications
dbspawn dbsrv11 demo
if [ $? != 0 ]; then
    echo Failed to start demo server
    exit
fi
# ok, now you can start the applications
...
```

3. 从 C 程序中生成守护程序，例如：

```
...
if( fork() == 0 ) {
    /* child process = start server daemon */
    execl( "/opt/sqlanywhere11/bin/dbsrv11",
"dbsrv11", "-ud", "demo" );
    exit(1);
}
/* parent process */
...
```

请注意，其中使用了 `-ud` 选项。

另请参见

- “[-ud 服务器选项](#)” 一节第 216 页
- “[后台启动服务器实用程序 \(dbspawn\)](#)” 一节第 781 页

了解 Windows 服务

可将数据库服务器作为 Microsoft Windows 程序而非服务来运行。但是，将数据库服务器作为标准程序运行以及在多用户环境中运行存在一些限制。

作为标准可执行文件来运行的限制

启动某个程序时，程序会在 Windows 登录会话中运行，这意味着如果用户注销该计算机，该程序会关闭。如果想要让某个程序多数时间都处于运行状态（通常对数据库服务器有这种要求），这种配置就会限制计算机的使用。为了使数据库服务器保持运行状态，用户必须在运行数据库服务器的计算机上保持登录状态。由于 Windows 计算机必须保持登录状态，因此这种配置还可能会带来安全风险。

服务的优点

将应用程序作为 Windows 服务安装的好处就是您注销登录之后它仍能运行。

启动某个服务时，该服务会使用一个叫作 LocalSystem 的特殊系统帐户（或用户指定的另一个帐户）登录。由于该服务不绑定到启动它的用户的用户 ID，则甚至当启动它的用户注销登录之后，该服务也保持打开状态。还可以将服务配置为在 Windows 计算机启动时于用户登录之前自动启动。

管理服务

在 SQL Anywhere 服务的管理上，Sybase Central 提供了一种比 Windows 服务管理器更方便、更全面的方法。也可以使用 dbsvc 实用程序来创建和修改服务。请参见“[用于 Windows 的服务实用程序 \(dbsvc\)](#)”一节第 773 页。

可作为 Windows 服务运行的程序

您可以将下列程序作为服务运行：

- 网络数据库服务器 (*dbsrv11.exe*)
- 个人数据库服务器 (*dbeng11.exe*)
- SQL Remote 消息代理 (*dbremote.exe*)
- 日志传送管理器实用程序 (*dbltn.exe*)
- MobiLink 服务器 (*mlsrv11.exe*)
- MobiLink 同步客户端 (*dbmlsync.exe*)
- MobiLink 中继服务器 (*rshost.exe*)
- MobiLink 中继服务器出站启动器 (*rsoe.exe*)
- MobiLink 监听器实用程序 (*dblsn.exe*)
- SQL Anywhere 广播转发器实用程序 (*dbns11.exe*)
- SQL Anywhere 卷影复制服务 (*dbvss11.exe*)

并非所有版本的 SQL Anywhere 都提供上述这些应用程序。

另请参见

- “[创建 Windows 服务](#)”一节第 60 页

管理 Windows 服务

可以通过命令行或 Sybase Central 中的 [服务] 选项卡执行以下 Windows 服务管理任务：

- 创建、编辑和删除服务
- 启动和停止服务
- 修改用于控制服务的参数
- 将数据库添加到服务中，以便可以同时运行几个数据库

Sybase Central 中的服务图标使用表示服务处于运行还是停止状态的图标来显示每个服务的当前状态。

创建 Windows 服务

本节将介绍如何使用 Sybase Central 和服务实用程序设置服务。

◆ 创建新服务（通过 Sybase Central）

1. 在左窗格中，选择 [SQL Anywhere 11]。
2. 在右窗格中，单击 [服务] 选项卡。
3. 选择 [文件] » [新建] » [服务]。
4. 请按照 [创建服务向导] 中的说明进行操作。

提示

也可以为 MobiLink 插件创建服务。请参见“在当前会话外运行 MobiLink 服务器”一节《MobiLink - 服务器管理》。

◆ 创建新服务（通过命令行）

- 使用 -w 选项运行 dbsvc 命令。

例如，要创建名为 myserv 的个人服务器服务并将数据库服务器作为 LocalSystem 用户运行，请输入以下命令：

```
dbsvc -as -w myserv "c:\Program Files\SQL Anywhere 11\bin32\dbeng11.exe"  
-n william -c 8m "c:\temp\sample.db"
```

请参见“用于 Windows 的服务实用程序 (dbsvc)”一节第 773 页。

注意

- 服务名称在前八个字符内必须唯一。
- 如果选择自动启动某项服务，则只要计算机启动 Windows，该服务便会启动。如果您选择手工启动服务，则您需要每次都从 Sybase Central 启动服务。如果要设置供将来使用的服务，则需要选中 [已禁用]。

- 在 Sybase Central 中创建服务时，在窗口中键入可执行文件的选项，但不必键入可执行文件名本身。例如，如果想要使用示例数据库运行高速缓存大小为 20 MB、名称为 myserver 的网络服务器，则要在 Sybase Central 中 [创建服务向导] 的 [参数] 文本框中键入以下内容：

```
-c 20M  
-n myserver samples-dir\demo.db
```

换行符可以使用，也可以不使用。

- 选择服务运行时将使用的帐户：特殊的 LocalSystem 帐户或另一个用户 ID。
有关此选择的详细信息，请参见“[设置帐户选项](#)”一节第 63 页。
- 如果希望能够从 Windows 桌面访问服务，请选择 [允许服务与桌面交互]。如果清除此选项，系统任务栏中不会出现任何图标，桌面上也不会出现任何窗口。

请参见“[配置 Windows 服务](#)”一节第 61 页。

删除 Windows 服务

删除服务时会从服务列表中将服务名删除。删除服务时并不会从硬盘中删除任何软件。

如果想要重新安装以前删除的服务，则需要重新键入选项。

◆ 删除 Windows 服务 (Sybase Central)

1. 在左窗格中，选择 [SQL Anywhere 11]。
在右窗格中，单击 [服务] 选项卡。
2. 在右窗格中，选择要删除的服务，然后从 [文件] 菜单中选择 [删除]。

◆ 删除 Windows 服务（通过命令行）

- 使用 -d 选项运行 dbsvc 实用程序。
例如，要在不提示您确认的情况下删除名为 myserv 的服务，请键入下面的命令：

```
dbsvc -y -d myserv
```

另请参见

- “[用于 Windows 的服务实用程序 \(dbsvc\)](#)”一节第 773 页

配置 Windows 服务

服务使用一组选项运行数据库服务器或其它应用程序。

除了选项以外，服务还接受用来指定服务在哪个帐户下运行以及服务的启动条件的其它参数。

◆ **更改服务的参数:**

1. 在左窗格中, 选择 [SQL Anywhere 11]。
2. 在右窗格中, 选择要更改的服务。
3. 从 [文件] 菜单中选择 [属性]。
4. 根据需要更改 [服务属性] 窗口的选项卡上的参数。
5. 完成后单击 [确定]。

对服务配置的更改将在用户下次启动服务时生效。[启动] 选项将在下次启动 Windows 时应用。

设置启动选项

以下选项用于控制 SQL Anywhere 服务的启动行为。可以在 [服务属性] 窗口的 [常规] 选项卡上设置它们。

- **自动** 如果选择 [自动] 设置, 则每当 Windows 操作系统启动时, 服务就会启动。此设置适用于数据库服务器和其它始终处于运行状态的应用程序。
- **手工** 如果选择 [手工] 设置, 则仅当具有管理员权限的用户启动服务时服务才会启动。有关管理员权限的信息, 请参见 Windows 文档。
- **已禁用** 如果选择 [已禁用] 设置, 服务将不会启动。

指定选项

服务的选项与可执行文件的选项相同。

小心

[服务属性] 窗口的 [配置] 选项卡包含一个 [参数] 文本框, 用于指定服务的选项。不要在此框中键入程序可执行文件的名称。

示例

要启动运行两个数据库、高速缓存大小为 20 MB、名称为 my_server 的网络服务器服务, 需要在 [参数] 字段中键入以下内容:

```
-c 20M  
-n my_server  
c:\db_1.db  
c:\db_2.db
```

要启动 [SQL Remote 消息代理] 服务, 并以用户 ID DBA 连接到示例数据库, 请键入以下内容:

```
-c "UID=DBA;PWD=sql;DBN=demo"
```

下图说明了示例 [服务属性] 窗口。



设置帐户选项

可以选择服务运行时使用的帐户。大多数服务都在特殊的 LocalSystem 帐户下运行，LocalSystem 帐户是服务的缺省选项。打开 [服务属性] 窗口中的 [帐户] 选项卡并键入另一帐户的信息，就可以将服务设置为使用该帐户登录。

如果选择使用 LocalSystem 以外的帐户运行服务，则该帐户必须拥有 [作为服务登录] 特权。可以通过 Windows 用户管理器应用程序的 [高级特权] 授予该特权。服务实用程序 (dbsvc) 也会在需要时授予此特权。

当图标出现在系统任务栏中时

- 如果服务以 LocalSystem 的名义运行，并且 [服务属性] 窗口中的 [允许服务与桌面交互] 处于选中状态，则在运行该服务的计算机上登录 Windows 的每个用户的桌面上都会出现一个图标。任何用户都可以打开应用程序窗口并停止作为服务运行的程序。

- 如果服务以 LocalSystem 的名义运行，而 [服务属性] 窗口中的 [允许服务与桌面交互] 未处于选中状态，则任何用户的桌面上都不会出现图标。只有拥有更改服务状态权限的用户才可以停止服务。
- 如果服务使用另一个帐户运行，则桌面上不会出现图标。只有拥有更改服务状态权限的用户才可以停止服务。

更改可执行文件

要更改与某个服务相关的程序可执行文件，请单击 [服务属性] 窗口中的 [配置] 选项卡，然后在 [文件名] 文本框中键入新路径和文件名。

如果将可执行文件移动到一个新的目录，您必须修改这一项。

向服务中添加新数据库

每个网络服务器或个人服务器都可以运行多个数据库。如果想要同时运行多个数据库，建议通过将新数据库附加到现有服务（而非创建新服务）来实现此目的。

◆ 向现有服务中添加新数据库

1. 在 [上下文] 下拉列表中，选择 [SQL Anywhere 11]。
2. 在右窗格中，单击 [服务] 选项卡。
3. 选择该服务，然后从 [文件] 菜单中选择 [属性]。
4. 单击 [配置] 选项卡。
5. 将新数据库的路径和文件名添加到 [参数] 框中选项列表的尾部。
6. 单击 [确定] 保存所作更改。

新数据库在下次启动服务时启动。

数据库可由客户端应用程序（如 Interactive SQL）在正在运行的服务器上启动。

有关从 Interactive SQL 启动数据库的详细信息，请参见“[START DATABASE 语句](#)”一节《[SQL Anywhere 服务器 - SQL 参考](#)》。

有关如何在嵌入式 SQL 应用程序中实现此功能的详细信息，请参见“[db_start_database 函数](#)”一节《[SQL Anywhere 服务器 - 编程](#)》。

从应用程序启动数据库时并不会将数据库附加到服务。如果停止后又重新启动了服务，则附加数据库将不会自动启动。

设置服务轮询频率

Sybase Central 可以按指定的间隔进行轮询以检查每个服务的状态（已启动或已停止），并更新图标以显示当前状态。在缺省情况下，轮询处于关闭状态。如果保持其关闭状态不变，则必须单击 **[刷新文件夹]** 才能查看对状态进行的更改。

◆ 设置 Sybase Central 轮询频率：

1. 在 **[上下文]** 下拉列表中，选择 **[SQL Anywhere 11]**。
2. 在右窗格中，单击 **[服务]** 选项卡。
3. 选择该服务，然后从 **[文件]** 菜单中选择 **[属性]**。
4. 单击 **[轮询]** 选项卡。
5. 选中 **[启用轮询]**。
6. 设置轮询频率。

该频率应用于所有服务，不仅仅是所选的服务。在此窗口中设置的值对后续会话仍然有效，有效性会持续到用户更改该值时为止。

7. 单击 **[确定]**。

启动和停止服务

◆ 启动或停止服务

1. 在 **[上下文]** 下拉列表中，选择 **[SQL Anywhere 11]**。
2. 在右窗格中，单击 **[服务]** 选项卡。
3. 选择该服务，然后从 **[文件]** 菜单中选择 **[启动]** 或 **[停止]**。

如果启动某个服务，则它会一直运行，直到将其停止。关闭 Sybase Central 或注销并不会停止该服务。

停止数据库服务器服务时会关闭到数据库的所有连接并停止数据库服务器。对其它应用程序而言，该程序已关闭。

Windows 服务管理器

可以使用 Sybase Central 来执行 SQL Anywhere 的所有服务管理。尽管可以使用 **[控制面板]** 中的 Windows **[服务管理器]** 来完成某些任务，但无法通过 Windows **[服务管理器]** 安装或配置 SQL Anywhere 服务。

如果打开 Windows **[服务管理器]**（位于 Windows **[控制面板]** 中），会出现一个服务列表。在用户于安装该服务时提供的服务名前面加上 SQL Anywhere，便构成了 SQL Anywhere 服务的名称。所有已安装的服务都会出现在该列表中。

服务依赖性

在某些情况下您可能需要运行多个可执行文件作为服务，而这些可执行文件可能相互依赖。例如，可能想要运行服务器和 SQL Remote 消息代理或日志传送管理器来协助进行复制。

在此类情况下，服务必须按正确的顺序启动。如果 SQL Remote 消息代理服务先于服务器启动，则该服务会因找不到服务器而启动失败。

可以通过使用**服务组**（通过 Sybase Central 进行管理）来防止出现这些问题。

服务组概述

可以将系统上的每个服务都指派为服务组的成员。缺省情况下，如下表所列，每个服务都属于某个组。

服务	缺省组
网络服务器	SQLANYServer
个人服务器	SQLANYEngine
MobiLink 同步客户端	SQLANYMLSync
复制代理	SQLANYLTM
SQL Remote 消息代理	SQLANYRemote
MobiLink 服务器	SQLANYMobiLink
广播转发器实用程序	SQLANYNS
MobiLink 监听器	SQLANYLSN
SQL Anywhere 卷影复制服务	SQLANYVSS

在对服务进行配置以确保其按正确的顺序启动之前，必须检查并确保该服务为某一适当组的成员。可以通过 Sybase Central 检查服务所属的组及更改此组。

◆ 检查并更改服务所属的组：

1. 在 [上下文] 下拉列表中，选择 [SQL Anywhere 11]。
2. 在右窗格中，单击 [服务] 选项卡。
3. 选择该服务，然后从 [文件] 菜单中选择 [属性]。
4. 单击 [依赖性] 选项卡。最上面的文本框显示该服务所属组的名称。
5. 单击 [更改] 以显示系统上可用组的列表。
6. 选择某个组或键入新组的名称。

7. 单击 **[确定]**，将服务指派到该组。

管理服务依赖性

可以使用 Sybase Central 指定服务的**依赖性**。例如：

- 您可以确保在服务组列表的每个组中至少有一个成员比当前服务先启动。
- 可以确保任意数量的服务先于当前服务启动。例如，可能想要确保先启动特定的网络服务器，然后再启动需要依赖该服务器才能运行的 SQL Remote 消息代理。

◆ 向依赖性列表中添加服务或组：

1. 在 **[上下文]** 列表中，选择 **[SQL Anywhere 11]**。
2. 在右窗格中，单击 **[服务]** 选项卡。
3. 选择服务，然后选择 **[文件] » [属性]**。
4. 单击 **[依赖性]** 选项卡。
5. 单击 **[添加服务]** 或 **[添加服务组]** 以便向依赖性列表中添加服务或组。
6. 从列表中选择一个服务或组。
7. 单击 **[确定]** 将服务或组添加到依赖性列表。

对服务器启动进行故障排除

本节将介绍在启动数据库服务器时可能发生的一些常见问题。

确保事务日志文件有效

如果现有事务日志无效，服务器将不会启动。例如，您可能在开发过程中将数据库文件替换为新版本，而却没有同时删除事务日志。然而，这会使事务日志文件与数据库不同，并会导致事务日志文件无效。

确保有足够的磁盘空间存放临时文件

SQL Anywhere 在运行时使用临时文件来存储信息。此文件一般存储在 SATMP 环境变量所指向的目录中，该目录通常是 `c:\temp`。

如果您没有足够的磁盘空间用于存放临时目录，将会在启动服务器时遇到问题。

请参见“SATMP 环境变量”一节第 357 页。

确保网络通信软件处于运行状态

在运行数据库服务器之前，必须已安装相应的网络通信软件且该软件已处于运行状态。如果您运行的是可靠的网络软件并且只安装了一个网络，此过程应该会比较简单。

有关网络通信问题的详细信息，请参见“客户端/服务器通信”第 133 页。

在运行数据库服务器之前，您应当确认其它需要网络通信的软件是否能正确运行。

如果是在 TCP/IP 协议下运行，则可能需要确认 ping 和 Telnet 的工作是否正常。许多 TCP/IP 协议栈都随附有 ping 和 Telnet 应用程序。

调试网络通信启动问题

如果在网络上建立连接时遇到问题，则在客户端和服务器均可以使用调试选项来诊断问题。在服务器上，使用 `-z` 选项。启动信息出现在数据库服务器消息窗口中：可以使用 `-o` 选项将结果记录到输出文件中。

请参见“`-z` 服务器选项”一节第 226 页和“`-o` 服务器选项”一节第 196 页。

确保使用的是正确的 `sasrv.ini` 文件

如果在网络上建立到服务器的连接时遇到问题，而所要连接的服务器是正确的，请尝试删除 `sasrv.ini` 文件。该文件包含服务器信息，包括服务器名、协议和地址。该文件中的服务器信息可能

会覆盖您在连接字符串中指定的信息。删除此文件会使 SQL Anywhere 创建一个新的 *sasrv.ini* 文件，其中包含用户在连接字符串中指定的信息。在 Windows 上，*sasrv.ini* 的缺省位置是 %ALLUSERSPROFILE%\Application Data\SQL Anywhere 11，而在 Unix 上为 ~/.sqlanywhere11。

如果在建立连接时仍然会遇到问题，则还应删除以下所有位置中 *sasrv.ini* 的任何副本：

- SQL Anywhere 安装目录的 *bin32* 或 *bin64* 子目录（在 *HKEY_LOCAL_MACHINE\SOFTWARE\Sybase\SQL Anywhere\11.0\Location* 注册表项中列出）
- Windows 目录
- Windows 系统目录
- 路径中的任何其它位置

有关 *sasrv.ini* 文件的详细信息，请参见“[高速缓存服务器名以提高连接速度](#)”一节第 129 页。

创建调试日志文件

可以使用 LogFile 连接参数来创建调试日志文件。日志文件可就连接发生故障的位置提供更多详细信息，从而帮助您排除故障并解决问题。请参见“[Logfile 连接参数 \[LOG\]](#)”一节第 273 页。

运行经过验证的 SQL Anywhere 应用程序

SQL Anywhere OEM 版是为 Sybase OEM 合作伙伴提供的。使用 SQL Anywhere OEM 版，经过验证的应用程序可对数据库执行任何操作，但受到授予该用户 ID 的权限的制约。

未经验证的连接具有只读访问权限，可对临时表执行插入、更新和删除操作。使用未经验证的连接将允许使用存储的过程创建复杂的报告，并可使用报告工具（例如 Crystal Reports）对其进行访问。

验证机制与任何应用程序编程语言或工具无关，可在每个连接上执行，从而使您在应用程序中可使用经过验证的连接和更多受限的未经验证的连接。

验证不是一种安全机制。对数据库运行未经验证的数据库服务器的任何用户可执行任何操作，但受到常见 SQL 权限模式的制约。

开发经过验证的应用程序

开发经过验证的应用程序是一个简单的过程：一个特殊的验证签名将并入到数据库中，第二个签名并入到应用程序中。当应用程序连接到数据库时，将比较两个签名以验证该应用程序。开发经过验证的 SQL Anywhere 应用程序需要执行以下步骤：

1. “获取验证签名”一节第 70 页
2. “验证数据库”一节第 71 页
3. “验证应用程序”一节第 71 页

包括在 SQL Anywhere 中的所有数据库工具（包括 Sybase Central、Interactive SQL 和 dbbackup 等实用程序）都是自验证的。它们对任何经过验证的数据库执行操作时将不会受到限制。如果数据库本身未经过验证，该工具将以受限、只读的方式执行操作。

对于经过验证的应用程序，必须使用 OEM 版的 SQL Anywhere 数据库服务器。此版本与普通数据库服务器的不同之处仅在于它可以处理验证指令。其它版本的数据库服务器忽略了验证指令。如果没有使用经过验证的数据库服务器，将不会对未经验证的应用程序施加限制。

获取验证签名

注意

要获得验证签名，必须与 Sybase iAnywhere 签订 OEM 合同。

◆ 获取验证签名

1. 访问 http://www.sybase.com/sql_anywhere_authentication_registration。
2. 完成表格以获取验证签名。以下信息将并入到您的验证机制中：
 - **公司** 公司的名称。
 - **应用程序名** 应用程序的名称。

有关公司名称和应用程序名如何并入验证机制中的信息，请参见“[验证数据库](#)”一节第 71 页。

完成该表格后，您将在 48 小时内通过电子邮件接收到一个数据库签名和一个应用程序签名。这些签名都是由字符和数字组成的长字符串（81 个字符）。包含您的验证信息的电子邮件消息中包括若干个关于如何使用这些信息的示例。某些电子邮件系统会在这些指令中强制添加换行符。确保重新连接电子邮件消息中已断开的行以使这些指令仍然有效。

验证数据库

SQL Anywhere OEM 版不允许对未经过验证的数据库执行任何操作。

可以使用 `Authenticated` 数据库属性来确定是否已经验证数据库：

```
SELECT DB_PROPERTY ( 'Authenticated' );
```

有关数据库属性的详细信息，请参见“[数据库属性](#)”一节第 601 页。

◆ 验证数据库

1. 使用以下 SQL 验证语句设置数据库的 `database_authentication` 选项：

```
SET OPTION PUBLIC.database_authentication =  
    'company = company-name;  
    application = application-name;  
    signature = database-signature';
```

2. `company-name` 和 `application-name` 参数是您获取签名时提供给 Sybase 的值，而 `database-signature` 是您从 Sybase 收到的数据库签名。
3. 重新启动数据库使该选项生效。

数据库服务器装载经过验证的数据库时，它在数据库服务器消息窗口中显示一条消息，说明经过验证的公司和应用程序。可以通过检查此消息是否存在来验证 `database_authentication` 选项是否生效。该消息具有以下格式：

```
此数据库授权用于:应用程序:application-name 公司:company-name
```

提示

可以将验证语句存储到 SQL 脚本文件中，以免还要键入长签名。可在 Interactive SQL 中运行 SQL 脚本，方法是在 [文件] 菜单中选择 [运行脚本]。

如果在 SQL Anywhere 安装目录的 `scripts` 子目录中创建了一个名为 `authenticate.sql` 的文件，并将验证语句存储到此文件中，则只要创建、重建或升级数据库，就会应用该文件。请参见“[升级经过验证的数据库](#)”一节第 74 页。

验证应用程序

连接后，经过验证的应用程序必须立即设置 `connection_authentication` 数据库选项。在连接建立后必须立即对每个连接设置该选项。ODBC 或 JDBC 应用程序查询数据库关于其功能的信息，而开发人

员可能无法控制这些操作。因此，在应用限制前每个连接都有一个三十秒的宽限期。在此宽限期内，无论要使用哪种开发工具，应用程序都可进行验证。

可以使用 **Authenticated** 连接属性来确定是否已经验证数据库：

```
SELECT CONNECTION_PROPERTY ( 'Authenticated' );
```

有关连接属性的详细信息，请参见“[连接属性](#)”一节第 566 页。

以下 SQL 语句验证连接：

```
SET TEMPORARY OPTION connection_authentication =  
'company = company-name;  
application = application-name;  
signature = application-signature';
```

只能使用 **TEMPORARY** 关键字为该连接的持续时间设置这一选项。*company-name* 和 *application-name* 必须与数据库验证语句中的相应项匹配。*application-signature* 是您从 Sybase 获取的签名。

数据库服务器将对照数据库签名验证该应用程序签名。如果对签名进行核实，则会验证连接，并且除了 SQL 权限强加的限制之外，连接的行为不受任何限制。如果未对签名进行核实，则连接的操作将仅限于未经验证的应用程序所允许的操作。

执行验证语句

执行用于设置验证选项的 **SET TEMPORARY OPTION** 语句的方式取决于您正使用的编程接口。此处所列签名为无效签名。所提供的示例使用以下接口设置验证选项：

- ODBC
- PowerBuilder
- JDBC
- ADO.NET
- 嵌入式 SQL

在验证选项中使用特殊字符

当公司名称中有引号、撇号或其它特殊字符（例如 Joe's Garage）时，您需要注意如何构造验证语句。整个验证选项集 (Company=...;Application=...;Signature=...) 是一个 SQL 字符串。在 SQL 的字符串规则中规定，如果字符串内包括引号，则它必须是双引号才能被接受。例如：

```
SET TEMPORARY OPTION connection_authentication=  
'Company = Joe''s Garage;  
Application = Joe''s Program;  
Signature = 0fa55157edb8e14d818e...';
```

ODBC

使用下面的语句：

```
SQLExecDirect (  
    hstmt,  
    "SET TEMPORARY OPTION connection_authentication=  
    'Company=MyCo;  
    Application=MyApp;
```

```

        Signature=0fa55159999e14d818e...';",
    SQL_NTS
);

```

必须在一行上输入该字符串，否则必须通过字符串连接来构建。

PowerBuilder

使用下面的 PowerScript 语句：

```

EXECUTE IMMEDIATE
    "SET TEMPORARY OPTION connection_authentication=
    'Company=MyCo;
    Application=MyApp;
    Signature=0fa551599998e14d818e...';"
USING SQLCA

```

JDBC

使用下面的语句：

```

Statement Stmt1 = con.createStatement();
Stmt1.executeUpdate(
    "SET TEMPORARY OPTION connection_authentication=
    'Company=MyCo;
    Application=MyApp;
    Signature=0fa55159999e14d818e...';"
);

```

必须在一行上输入该字符串，否则必须通过字符串连接来构建。

ADO.NET

使用下面的语句：

```

SACommand cmd=new SACommand(
    "SET TEMPORARY OPTION connection_authentication=
    'Company=MyCo;
    Application=MyApp;
    Signature=0fa551599998e14d818e...';",
    con
);
cmd.ExecuteNonQuery();

```

必须在一行上输入该字符串，否则必须通过字符串连接来构建。

嵌入式 SQL

```

Use the following statement:
EXEC SQL SET TEMPORARY OPTION connection_authentication=
    'Company=MyCo;
    Application=MyApp;
    Signature=0fa551599998e14d818e...';

```

必须在一行上输入该字符串，否则必须通过字符串连接来构建。

连接到经过验证的数据库时，应分别执行连接和验证步骤。但是某些对象（例如 Visual Basic Grid 对象）可尝试独立的隐式连接，该连接不会自动包括验证。在这种情况下，不会对该连接进行验证，而数据库操作会失败。可通过在连接字符串中包括 `InitString` 连接参数来避免此问题。下面的示例说明了如何修改 Visual Basic 应用程序以包括 `InitString` 连接参数，以便在每个连接建立后立即进行验证：

```
mConnectionString =
"Provider=SAPROV.11;
UID=DBA;
PWD=sql;
ENG=test11;
InitString=SET TEMPORARY OPTION connection_authentication=
'Company=MyCo;
Application=MyApp;
Signature=0fa55157edb8e14d818e...'"
mdbName.ConnectionString = mConnectionString
mdbName.Open
mIsSQL = True
```

升级经过验证的数据库

升级或重建数据库时，能够保留验证信息的唯一方法是将验证语句存储在文件 *authenticate.sql* 中。

升级实用程序不能将数据库升级到版本 11 或更高

无法使用升级实用程序 (dbupgrad) 将 9.0.2 或更早版本的数据库升级到版本 10 或更高。要将较旧版本的数据库升级到版本 11 或更高，您必须重建数据库，方法是先执行卸载然后重装数据库。请参见“升级 SQL Anywhere”一节《SQL Anywhere 11 - 更改和升级》。

在 *install-dir\scripts* 目录中创建名为 *authenticate.sql* 的文件，其中包含以下内容：

```
SET OPTION PUBLIC.database_authentication = 'authentication-statement'
go
```

文件中必须有 `go`；否则，将忽略该语句。

有关字符串 *authentication-statement* 内容的信息，请参见“[database_authentication \[数据库\]](#)”一节第 495 页。

运行 SQL Anywhere Web 版应用程序

SQL Anywhere Web 版是 SQL Anywhere 的免费版本，可供开发和部署 Web 应用程序使用。SQL Anywhere Web 版只能用于 Web 浏览器应用程序，并可在 Windows 和 Linux 上运行。在数据库大小、高速缓存大小、CPU、优化技术、执行策略或 SQL 语言支持方面没有任何限制。

某些功能不可用于使用 SQL Anywhere Web 版的应用程序，并要求您以单独授权组件的形式来购买这些功能，或升级为付费 SQL Anywhere 许可。

有关 SQL Anywhere Web 版的详细信息（包括授权信息、功能可用性信息和平台支持详细信息），请参见 SQL Anywhere Web 版常见问题解答，网址为 <http://www.sybase.com/detail?id=1057560>。

SQL Anywhere 中的错误报告

如果发生了致命错误或崩溃，并且以下任一应用程序检测到了这种情况，则会创建一个错误报告，说明问题出现时的情形：

- Interactive SQL (dbisql)
- MobiLink 监听器 (dbsln)
- MobiLink 服务器 (mlsrv11)
- 网络服务器 (dbsrv11)
- 个人服务器 (dbeng11)
- QAnywhere 代理 (qaagent)
- 复制代理 (dbltn)
- 用于 MobiLink 的 SQL Anywhere 客户端 (dbmlsync)
- SQL Anywhere 控制台实用程序 (dbconsole)
- SQL Remote (dbremote)
- Sybase Central

错误报告包括诸如发生崩溃时线程执行状态等信息，以便 iAnywhere 能够更有效地诊断问题的成因。缺省情况下，将在诊断目录（由 SADIAGDIR 环境变量指定）中创建错误报告，如果此位置不存在，将在数据库文件所在的目录中创建错误报告。

错误报告文件名由以下部分组成：

- 标识应用程序的前缀：

应用程序标识符	应用程序
LSN	监听器实用程序
LTM	复制代理
MLC	MobiLink 客户端
MLS	MobiLink 服务器
QAA	QAnywhere 代理
SA	个人或网络数据库服务器
SR	SQL Remote

- 表示软件版本的值
- 以下划线链接的两个字段，提供错误报告创建时间的时间戳
- 应用程序标识符
- *.mini_core* 扩展名

例如, `SAll_20051220_133828_32116.mini_core` 是一个来自 SQL Anywhere 版本 11 数据库服务器的错误报告, 日期为 2006/06/20, 时间为 1:38:28 pm, 发生错误的进程是 32116。

数据库服务器常规运行期间, 还会记录有关数据库服务器的诊断信息, 如计算机上安装的 CPU 数量、是否启用了超线程以及启动服务器时指定的选项。也可以使用 `dbsupport` 来提交此信息。

SQL Anywhere 软件如何提交错误报告和诊断信息

数据库服务器成功写出错误报告信息后, 会启动支持实用程序 (`dbsupport`) 并将要提交的错误报告文件的名称传递给它。缺省情况下, `dbsupport` 会尝试提示用户在生成错误报告时便提交该报告, 但如果 `dbsupport` 无法提示用户, 则不会发送报告。`iAnywhere` 鼓励用户在错误报告生成时便将其提交。报告不包含任何表明发送者身份的信息。

错误报告和诊断信息通过 HTTP 上载到 `iAnywhere` 错误报告 Web 站点。此过程节省了您的时间, 即通过尽可能方便的方式将相关文件发送给 `iAnywhere`, 以便可以对用户遇到的问题进行诊断并提供相应的解决方案。

可以通过 `-cc` 选项更改 `dbsupport` 的缺省行为:

- 以下命令会将 `dbsupport` 配置为不提示用户而自动提交错误报告:

```
dbsupport -cc autosubmit
```

- 以下命令会禁用错误报告自动提交功能:

```
dbsupport -cc no
```

如果选择不提交错误报告, 该报告将保留在硬盘上的诊断目录中。诊断目录的位置由 `SADIAGDIR` 环境变量指定。请参见“[SADIAGDIR 环境变量](#)”一节第 353 页。

可以通过 `-lc` 选项查看错误报告的列表:

- 下面的命令生成一份所有尚未提交给 `iAnywhere Solutions` 的崩溃报告的列表:

```
dbsupport -lc
```

将错误报告提交给 `iAnywhere` 可以协助诊断致命错误或声明的成因。错误报告提交后, 系统即会从生成它的计算机上将其删除。请参见“[支持实用程序 \(dbsupport\)](#)”一节第 785 页。

可以通过 `-sc` 选项手动提交错误报告:

- 下面命令将所有存储于诊断目录中的崩溃报告和诊断信息提交给 `iAnywhere Solutions`:

```
dbsupport -sa
```

SQL Anywhere 数据库连接

目录

连接参数	80
通过 SQL Anywhere API 连接	83
从桌面应用程序连接到 Windows Mobile 数据库	85
通过 Sybase Central、Interactive SQL 或 SQL Anywhere 控制台实用程序进行连接	86
创建 ODBC 数据源	90
使用 OLE DB 连接到数据库	97
使用集成登录	99
Kerberos 验证	107
示例 SQL Anywhere 数据库连接	116
对连接进行故障排除	124
从数据库断开连接	132

客户端应用程序的所有活动都必须借助数据库连接才可以进行。直到建立了**连接**，客户端应用程序才能与数据库服务器进行交互。当数据库服务器连接建立后，用户 ID 将决定用户有权在数据库服务器上执行哪些操作。

用户连接到数据库时，数据库服务器会为该用户的连接指派唯一的**连接 ID**。对于数据库服务器的每个新连接，服务器将连接 ID 值递增 1。连接 ID 记录在 -z 服务器输出中。连接 ID 可用于过滤请求记录信息，标识哪个连接在数据库上放置了锁，或跟踪服务器启动后与服务器之间存在的连接总数以及这些连接的先后顺序。请参见“请求记录”一节《SQL Anywhere 服务器 - SQL 的用法》和“锁定的工作方式”一节《SQL Anywhere 服务器 - SQL 的用法》。

可以使用 CONNECTION_PROPERTY 函数来获得用户的 *connection-id*。请参见“CONNECTION_PROPERTY 函数 [System]”一节《SQL Anywhere 服务器 - SQL 参考》。

连接参数

当连接到数据库时，应用程序使用一组**连接参数**来定义连接。连接参数可包括服务器名、数据库名和用户 ID。

使用连接参数时，完成某个任务有多种方法。例如，您可以使用 `DatabaseName (DBN)` 连接参数（推荐使用）或 `DatabaseSwitches (DBS)` 参数来启动嵌入式数据库。

每个连接参数指定一个关键字-值对（采用 `parameter=value` 的形式）。以下示例将在口令连接参数中使用缺省口令：

```
Password=sql
```

连接参数汇编到**连接字符串**中。连接字符串中用分号分隔各个连接参数：

```
ServerName=demo11;DatabaseName=demo
```

表示连接字符串

连接字符串示例可以用以下形式表示：

```
parameter1=value1  
parameter2=value2  
...
```

这等同于以下连接字符串：

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

必须在一行中输入一个连接字符串，并用分号分隔各个参数设置。

连接参数的语法规则

- **包含空格的连接字符串** 如果任何连接参数值中包含空格，必须用双引号将整个连接字符串括起来。
- **布尔值** 布尔（真或假）参数在条件为真时值为 YES、ON、1、TRUE、Y 或 T，在条件为假时，值为 NO、OFF、0、FALSE、N 或 F。
- **区分大小写** 虽然连接参数的值可能区分大小写（例如 Unix 的文件名），但连接参数不区分大小写。

可按优先级顺序从以下位置获取由接口库使用的连接参数：

- **连接字符串** 您可以在连接字符串中显式传递参数。
- **SQLCONNECT 环境变量** SQLCONNECT 环境变量可存储连接参数。
- **数据源** ODBC 数据源可存储参数。
- **字符集限制** 建议服务器名必须由 ASCII 字符集中 1 到 127 范围内的字符组成。其它参数没有此限制。
- **优先级** 以下规则控制参数的优先级：

- 连接字符串中的条目按从左到右的顺序读取。如果对同一个参数指定了多次，则使用字符串中的最后一个。在这一点上 ODBC、OLE DB、Sybase Central、Interactive SQL 和 SQL Anywhere 控制台实用程序是例外：如果多次指定同一参数，则将应用第一个字符串。
- 如果字符串包含数据源或文件数据源条目，则从配置文件中读取有关信息，如果尚未设置该文件中的条目，则使用这些条目。例如，如果连接字符串包含一个数据源名称，并且还显式设置了该数据源中包含的一些参数，则在发生冲突时使用显式参数。
- **连接字符串分析** 如果分析连接字符串时出现问题，将产生一个错误，该错误会指出导致问题出现的连接参数。
- **空连接参数** 如果连接参数被指定为空值，该空值将被视为长度为零的字符串。

另请参见

- [“连接参数和网络协议选项”第 247 页](#)
- [“连接字符串和字符集”一节第 386 页](#)

以连接字符串形式传递的连接参数

连接参数以**连接字符串**的形式传递到接口库。该字符串由一组用分号隔开的参数组成：

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

通常，由应用程序构建并传递到接口库的连接字符串并不直接与用户输入信息的方式相对应。相反，用户可以完成窗口，或者应用程序可以从初始化文件中读取连接信息。

许多 SQL Anywhere 实用程序以 `-c` 选项的形式接受连接字符串，并且不加更改便将该连接字符串传递到接口库。以下示例是一个典型的备份实用程序 (dbbackup) 的命令行：

```
dbbackup -c "ENG=sample_server;DBN=demo;UID=DBA;PWD=sql" SQLAnybackup
```

另请参见

- [“解决连接参数冲突”一节第 81 页](#)

解决连接参数冲突

要解决连接参数冲突：

- **使用 DBF 指定数据库文件** 使用 StartLine (START) 参数或 DatabaseFile (DBF) 连接参数（推荐使用）来指定数据库文件。
- **使用 DBN 指定数据库名** 使用 StartLine (START) 参数、DatabaseSwitches (DBS) 连接参数或 DatabaseName (DBN) 连接参数（推荐使用）来指定数据库名。
- **使用 ENG 指定数据库服务器名** 自动启动一个尚未运行的数据库文件时，请在 ServerName (ENG) 参数中指定数据库服务器的名称。这可确保数据库连接到目标数据库服务器。
- **使用 Start 参数指定高速缓存大小** 使用 StartLine (START) 连接参数调整 DatabaseFile (DBF) 连接参数启动数据库文件的方式。

例如，以下的嵌入式数据库连接参数将启动一个具有更多高速缓存的数据库服务器：

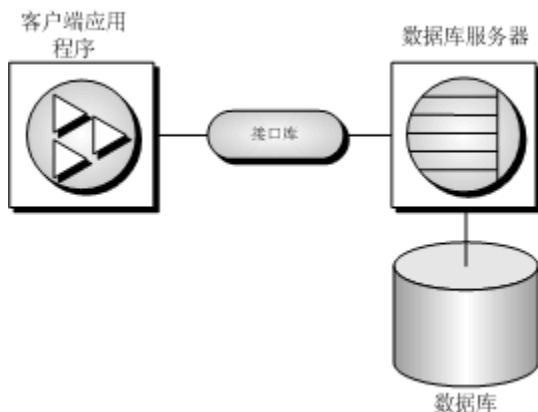
```
DBF=samples-dir\demo.db
DBN=Sample
ENG=Sample Server
UID=DBA
PWD=sql
START=dbeng11 -c 8M
```

通过 SQL Anywhere API 连接

要连接到数据库，客户端应用程序必须调用以下 SQL Anywhere API 函数之一：

接口	详细信息
ODBC	“SQL Anywhere ODBC API” 《SQL Anywhere 服务器 - 编程》 “创建 ODBC 数据源” 一节第 90 页
OLE DB	“使用 OLE DB 连接到数据库” 一节第 97 页
ADO.NET	“连接到数据库” 一节 《SQL Anywhere 服务器 - 编程》
嵌入式 SQL	“SQL Anywhere 嵌入式 SQL” 《SQL Anywhere 服务器 - 编程》
Sybase Open Client	“将 SQL Anywhere 用作 Open Server” 第 1057 页 “Sybase Open Client API” 《SQL Anywhere 服务器 - 编程》
iAnywhere JDBC 驱动程序	“从 JDBC 客户端应用程序连接” 一节 《SQL Anywhere 服务器 - 编程》 “SQL Anywhere JDBC 驱动程序” 《SQL Anywhere 服务器 - 编程》
jConnect JDBC 驱动程序	“从 JDBC 客户端应用程序连接” 一节 《SQL Anywhere 服务器 - 编程》 “SQL Anywhere JDBC 驱动程序” 《SQL Anywhere 服务器 - 编程》

SQL Anywhere API 使用客户端应用程序的调用中包含的连接信息来查找数据库服务器，并与之连接。由客户端应用程序发送的信息还包括保存在数据源、SQLCONNECT 环境变量或服务器地址高速缓存中的信息。下图是该过程的简单表示。



其它信息

如果需要……	请考虑阅读……
从 Sybase Central 或 Interactive SQL 进行连接的概述（包括涉及到的驱动程序的说明）	“通过 Sybase Central、Interactive SQL 或 SQL Anywhere 控制台实用程序进行连接”一节第 86 页
快速入门的一些示例（包括 Sybase Central 和 Interactive SQL 的一些使用场合）	“示例 SQL Anywhere 数据库连接”一节第 116 页
了解数据源	“创建 ODBC 数据源”一节第 90 页
了解有哪些连接参数可供使用	“连接参数”一节第 248 页
查看如何建立连接的深入说明	“对连接进行故障排除”一节第 124 页
了解网络特定的连接问题	“客户端/服务器通信”第 133 页
了解影响连接的字符集问题	“连接字符串和字符集”一节第 386 页
了解穿过防火墙进行连接	“穿过防火墙进行连接”一节第 136 页

从桌面应用程序连接到 Windows Mobile 数据库

您可以通过在台式机上运行的应用程序（如 Sybase Central 或 Interactive SQL）连接到在 Windows Mobile 设备上运行的数据库服务器。该连接在台式机和 Windows Mobile 设备之间的 ActiveSync 链接上使用 TCP/IP。

另请参见

- “在 Windows Mobile 设备上启动数据库服务器” 一节第 318 页
- “创建 ODBC 数据源以连接到 Windows Mobile 设备” 一节第 319 页
- “确定 Windows Mobile 设备的 IP 地址” 一节第 318 页

通过 Sybase Central、Interactive SQL 或 SQL Anywhere 控制台实用程序进行连接

本节提供 [连接] 窗口的使用过程。在 Sybase Central、Interactive SQL 和 SQL Anywhere 控制台实用程序中，使用 [连接] 窗口来定义数据库服务器连接参数。

另请参见

- “示例 SQL Anywhere 数据库连接” 一节第 116 页

使用 [连接] 窗口

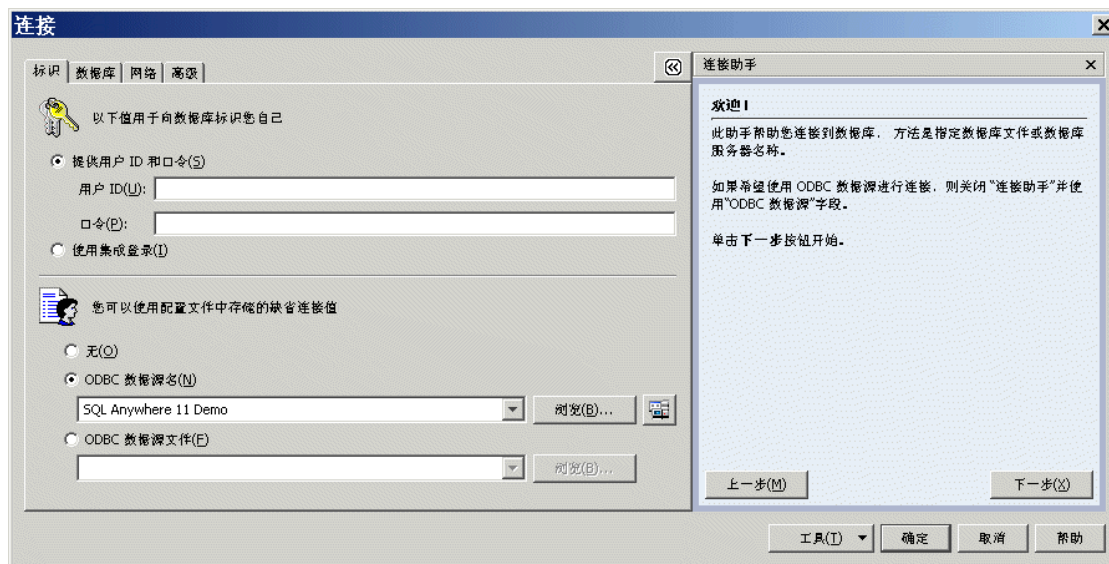
当从 Sybase Central、Interactive SQL 或 SQL Anywhere 控制台实用程序连接到服务器或数据库时，应使用 [连接] 窗口来定义连接参数。在 [连接] 窗口中输入的信息在两个会话之间不予保留。

在 [连接] 窗口中指定的连接参数取决于数据库服务器上正在运行的数据库数目。要连接到单个数据库，请填写 [用户 ID] 和 [口令] 字段。如果有多个数据库正在数据库服务器上运行，则必须指定其它连接参数，例如服务器或数据库的名称。

[连接] 窗口中包含以下内容：

- [标识] 选项卡。可使用该选项卡指定您的用户名、口令和数据源。
- [数据库] 选项卡。可使用该选项卡标识要连接的服务器或数据库。
- [高级] 选项卡。可使用该选项卡指定其它连接参数以及为连接指定驱动程序。

[连接] 窗口包含一个 [连接助手]，可以帮助您连接到数据库。要显示或隐藏 [连接助手]，可单击窗口右上角的箭头。



单击 [工具] 访问以下工具：

- [测试连接] 工具用于在您退出 [连接] 窗口前测试连接。
- [将连接字符串复制到剪贴板] 工具用于通过您在 [连接] 窗口中指定的选项创建连接字符串并将该字符串复制到剪贴板。
- [另存为 ODBC 数据源] 工具用于通过指定的选项快速创建 ODBC 数据源。

在成功连接到数据库后，数据库名称会出现在 Sybase Central 的 [文件夹] 窗格中运行该数据库的数据库服务器名称的下方。该连接的用户 ID 会出现在数据库名称之后。

在 Interactive SQL 中，数据库名、用户 ID 和数据库服务器名会出现在标题栏中。

打开 [连接] 窗口

在启动 Sybase Central 时，需要手动打开 [连接] 窗口。

在启动 Interactive SQL 时，[连接] 窗口会自动出现。要手动打开它，请选择 [SQL] » [连接]。

◆ 打开 [连接] 窗口 (Sybase Central)

- 在 Sybase Central 中，选择 [连接] » [使用 SQL Anywhere 11 连接]。或者，按 F11 键打开 [连接] 菜单。

提示

您可以使用连接配置文件更方便快速地与指定数据库建立后续连接。

◆ 打开 [连接] 窗口 (Interactive SQL)

1. 在 Interactive SQL 中，选择 [SQL] » [连接]。
或者，您可以按 F11 打开 [连接] 窗口。
2. 指定数据库的连接参数。例如，要连接到示例数据库：
 - 单击 [标识] 选项卡。
 - 单击 [ODBC 数据源名称]，然后选择 [SQL Anywhere 11 Demo]。
 - 单击 [OK]。

◆ 打开 [连接] 窗口 (SQL Anywhere 控制台实用程序)

- 运行以下命令：

```
dbconsole
```

Sybase Central 连接配置文件

首次连接到数据库服务器或数据库时，应输入用户 ID、口令和其它连接参数。在进行后续连接时，必须再次输入这些信息。为了节省时间和简化连接过程，可创建连接配置文件以保存每个数据库的连接参数。

要使用和管理连接配置文件，请选择 **[连接] » [连接配置文件]**。此命令会打开 **[连接配置文件]** 窗口，可在其中执行以下操作：

- 使用连接配置文件进行连接
- 编辑现有的连接配置文件
- 创建新的连接配置文件
- 设置配置文件的说明
- 删除或移除配置文件
- 导入或导出连接配置文件
- 设置一个配置文件以在 Sybase Central 启动时自动连接

注意

连接配置文件是 Sybase Central 特有的。如果您正在建立一个 ODBC 应用程序，则可以使用 ODBC 数据源实现与连接配置文件相似的功能。请参见“[创建 ODBC 数据源](#)”一节第 90 页。

创建连接配置文件

◆ 创建新的连接配置文件

1. 在 Sybase Central 中，选择 **[连接] » [连接配置文件]**。
2. 单击 **[新建]**。
3. 在 **[名称]** 字段中，键入新配置文件的名称。
4. 选择 **[新建连接配置文件]** 并从列表中选择适当的插件。插件为产品，例如 **SQL Anywhere 11** 或 **MobiLink 11**。

要基于现有配置文件创建新的连接配置文件，请选择 **[复制连接配置文件]**，并从 **[现有连接配置文件]** 列表中选择配置文件。

5. 要允许其他用户访问该配置文件，请选择 **[与其他用户共享此连接配置文件]**。该设置在多用户平台（例如 Unix）上比较有用。
6. 单击 **[确定]**。
7. 在 **[编辑连接配置文件]** 窗口中输入所需的值，然后单击 **[确定]** 关闭窗口。

◆ **在 Sybase Central 启动时自动连接**

1. 在 Sybase Central 中，选择 [连接] » [连接配置文件]。
2. 在 [连接配置文件] 列表中选择一個连接配置文件。
3. 单击 [设置启动]，将 [启动时使用] 列从 [否] 更改为 [是]。

编辑连接配置文件

◆ **编辑现有连接配置文件的参数**

1. 在 Sybase Central 中，选择 [连接] » [连接配置文件]。
2. 在 [连接配置文件] 列表中选择一個连接配置文件。
3. 单击 [编辑]。
4. 在 [编辑连接配置文件] 窗口中编辑值。

导入连接配置文件

◆ **导入连接配置文件**

1. 在 Sybase Central 中，选择 [连接] » [连接配置文件]。
2. 单击 [导入]。
3. 在 [文件名] 字段中键入所要导入的连接配置文件的名称。
4. 单击 [确定]。

导出连接配置文件

◆ **导出连接配置文件**

1. 在 Sybase Central 中，选择 [连接] » [连接配置文件]。
2. 在 [连接配置文件] 列表中选择一個连接配置文件。
3. 单击 [导出]。
4. 在 [文件名] 字段中，键入连接配置文件的文件名。
5. 单击 [保存]。

创建 ODBC 数据源

Microsoft 的**开放式数据库连接 (ODBC)** 是连接客户端应用程序与基于 Windows 的数据库管理系统的标准应用程序编程接口。

许多客户端应用程序（包括应用程序开发系统）使用 ODBC 接口访问 SQL Anywhere。当连接到数据库时，ODBC 应用程序通常使用 ODBC 数据源。ODBC 数据源是一组存储在注册表或文件中的连接参数。

小心

不建议在数据源中存储用户 ID、加密或未加密口令以及数据库密钥。

SQL Anywhere ODBC 驱动程序名为 *dbodbc11.dll*，位于 *install-dir\bin32*。

有关在 SQL Anywhere 中使用 ODBC 的详细信息，请参见“对 ODBC 的支持”一节《SQL Anywhere 服务器 - 编程》。

可以使用 ODBC 数据源从以下应用程序连接到 SQL Anywhere 数据库：

- Sybase Central、Interactive SQL 和 SQL Anywhere 控制台实用程序。
- 所有 SQL Anywhere 实用程序。
- PowerDesigner 物理数据模型和 InfoMaker。
- 支持 ODBC 的任何应用程序开发环境，例如 Microsoft Visual Basic、Sybase PowerBuilder 和 Borland Delphi。
- Unix 上的 SQL Anywhere 客户端应用程序。在 Unix 上，数据源作为文件存储。

存储 SQL Anywhere 连接参数

您可以使用 ODBC 数据源连接到 ODBC 数据库。客户端计算机要求每个数据库连接都具有一个 ODBC 数据源。

ODBC 数据源包含一组连接参数。您可以将多组 SQL Anywhere 连接参数以 ODBC 数据源的形式存储在 Windows 注册表中，或以文件形式存储。

对于 SQL Anywhere，ODBC 数据源的使用范围超出了使用 ODBC 接口的 Windows 应用程序：

- UNIX 和 Windows 操作系统上的 SQL Anywhere 客户端应用程序都可以使用 ODBC 数据源。
- 所有 SQL Anywhere 客户端接口（jConnect 和 Open Client 除外）都可以使用 ODBC 数据源。在 Unix 和 Windows Mobile 操作系统上，数据源存储在文件中。

一旦有了数据源，您的连接字符串就可以指明要使用的数据源：

- **数据源** 使用 DataSourceName (DSN) 连接参数引用 Windows 注册表中的数据源：

`DSN=my-data-source`

- **文件数据源** 使用 FileDataSourceName (FILEDSN) 连接参数引用保存在文件中的数据源：

```
FileDSN=mysource.dsn
```

注意

在创建连接字符串时，它可以包含含有连接参数的 ODBC 数据源名称和明确指定的连接参数。如果在连接字符串和 ODBC 数据源内同时指定了连接参数，则明确指定的值优先。

使用 [连接] 窗口创建 ODBC 数据源

可以在 Sybase Central、Interactive SQL 和 SQL Anywhere 控制台实用程序中使用 [连接] 窗口来创建 ODBC 数据源。

◆ 使用 [连接] 窗口创建 ODBC 数据源

1. 打开 [连接] 窗口。请参见“打开 [连接] 窗口”一节第 87 页。
2. 指定 [用户 ID]、[口令] 和 [文件名]。
3. 选择 [工具] » [另存为 ODBC 数据源]。
4. 在 [输入此新数据源的名称] 字段中，键入数据源的名称。
5. 在 [选择数据源类型] 列表中，指定此数据源是只对当前用户有效还是对所有用户均有效。
6. 单击 [保存]。
7. 单击 [确定]。

使用 [ODBC 管理器] 创建 ODBC 数据源

在基于 Windows 的应用程序上，可以使用 Microsoft 的 [ODBC 管理器] 创建和编辑数据源。可使用该实用程序处理 [用户数据源]、[文件数据源] 和 [系统数据源]。

小心

不建议在数据源中存储用户 ID、加密或未加密口令以及数据库密钥。

◆ 创建 ODBC 数据源（ODBC 管理器）：

1. 选择 [开始] » [程序] » [SQL Anywhere 11] » [ODBC 管理器]。
2. 要为当前用户创建 ODBC 数据源，请单击 [用户 DSN] 选项卡。
要创建系统范围的 ODBC 数据源，请单击 [系统 DSN] 选项卡。
3. 单击 [添加]。
4. 在 [名称] 列表中，选择 [SQL Anywhere 11]。单击 [完成]。
5. 指定 ODBC 数据源的连接参数。
6. 单击 [确定]。

7. 单击 [确定]。

在 64 位 Windows 上创建系统 ODBC 数据源

64 位版本的 Windows 维护两组 [系统数据源] 集合，其中一组适用于 64 位应用程序，另一组适用于 32 位应用程序。要创建 64 位应用程序和 32 位应用程序都可以访问的系统数据源，必须运行 32 位 ODBC 管理器的副本（位于 *WINDOWS\SysWOW64* 文件夹）。要避免出现连接问题，请将 32 位系统数据源设置得与 64 位系统数据源完全相同。

◆ 使用 [ODBC 管理器] 编辑 ODBC 数据源

1. 选择 [开始] » [程序] » [SQL Anywhere 11] » [ODBC 管理器]。
2. 单击 [用户 DSN] 选项卡。
3. 在 [名称] 列表中单击某个数据源。
4. 单击 [配置]。
5. 编辑 ODBC 数据源的连接参数。
6. 单击 [确定]。
7. 单击 [确定]。

使用 dbdsn 实用程序创建 ODBC 数据源

无法通过 dbdsn 实用程序创建文件数据源。可使用 [ODBC 管理器] 创建文件数据源。系统数据源仅限于基于 Windows 的操作系统。

小心

不建议在数据源中存储用户 ID、加密或未加密口令以及数据库密钥。

◆ 创建 ODBC 数据源（命令行）：

- 运行 dbdsn 命令，指定要使用的连接参数。

例如，以下命令用于为示例数据库创建数据源。该命令必须在一行中输入：

```
dbdsn -w "My DSN" -c "UID=DBA;PWD=sql;DBF=samples-dir\demo.db"
```

有关 *samples-dir* 的信息，请参见“[示例目录](#)”一节第 368 页。

有关 dbdsn 实用程序的详细信息，请参见“[数据源实用程序 \(dbdsn\)](#)”一节第 710 页。

在 64 位 Windows 上创建系统 ODBC 数据源

64 位版本的 Windows 维护两组 [系统数据源] 集合，其中一组适用于 64 位应用程序，另一组适用于 32 位应用程序。要创建 64 位应用程序和 32 位应用程序都可以访问的系统数据源，必须运行 32 位版本的 dbdsn（位于 *SQL Anywhere bin32* 文件夹）。要避免出现连接问题，请将 32 位系统数据源设置得与 64 位系统数据源完全相同。

在 Mac OS X 上创建 ODBC 数据源

在创建 ODBC 数据源之前，必须先添加 SQL Anywhere ODBC 驱动程序。

小心

不建议在数据源中存储用户 ID、加密或未加密口令以及数据库密钥。

◆ 添加 SQL Anywhere ODBC 驱动程序

1. 从 */Applications/Utilities* 启动 ODBC 管理器。
2. 选择 **[Drivers]** 选项卡。
3. 单击 **[Add]**。
4. 在 **[Description]** 字段中键入 **SQL Anywhere 11**。
5. 单击 **[Choose]**，然后在 **[Driver File Name]** 和 **[Setup File Name]** 两个字段中选择 SQL Anywhere ODBC 驱动程序。缺省情况下，它位于 */Applications/SQLAnywhere11/System/lib/dbodbc11_r.bundle*。

软件包名称中的 *_r* 表明该软件包为驱动程序的线程版本。还有一个非线程版本 (*dbodbc11.bundle*)，用于非线程应用程序。

6. 单击 **[OK]**。

◆ 创建 ODBC 数据源

可使用文本编辑器添加信息。ODBC 配置文件位于主目录的 */Library/ODBC* 中。其中，*odbcinst.ini* 文件存储驱动程序信息，*odbc.ini* 文件存储数据源信息。

还可以使用数据源实用程序 (dbdsn) 在 Mac OS X 上创建 ODBC 数据源。请参见“[数据源实用程序 \(dbdsn\)](#)”一节第 710 页。

1. 从 */Applications/Utilities* 启动 ODBC 管理器。
2. 在 [ODBC Administrator] 中，单击 **[User DSN]** 选项卡，然后单击 **[Add]**。
3. 在 **[名称]** 列表中，单击 **[SQL Anywhere 11]**。
4. 单击 **[完成]**。
5. 在 **[数据源名]** 字段中，键入 **Demo11**。
6. 添加以下连接参数。这些连接参数和值不区分大小写。

关键字	值
用户 ID	DBA
口令	sql
Start Line	dbeng11

关键字	值
数据库文件	<i>/Applications/SQLAnywhere11/System/demo.db</i>
ThreadManager	ON
Driver	SQL Anywhere 11

有关连接参数的详细信息，请参见“[连接参数和网络协议选项](#)”第 247 页。

7. 单击 [OK]。
8. 单击 [Apply]。
9. 按 Command+Q 退出 ODBC 管理器。

在 Windows 上使用文件数据源

通常，在基于 Windows 的操作系统上，ODBC 数据源存储在系统注册表中。文件数据源是一种以文件形式存储的替代形式。在 Windows 中，文件数据源的扩展名通常为 *.dsn*。它们由不同部分组成，每一部分都以包含在方括号中的名称开始。

要使用文件数据源进行连接，请使用 FileDataSourceName (FILEDSN) 连接参数。不能在同一连接中同时使用 DataSourceName (DSN) 和 FileDataSourceName (FILEDSN)。

可发布的文件数据源

可使用文件数据源将文件分发给用户，并简化对多用户连接的管理。如果文件放在文件数据源的缺省位置，则它由 ODBC 自动获取。

◆ 创建 ODBC 文件数据源（ODBC 管理器）：

1. 选择 [开始] » [程序] » [SQL Anywhere 11] » [ODBC 管理器]。
2. 单击 [文件 DSN] 选项卡。
3. 单击 [添加]。
4. 在 [名称] 列表中，单击 [SQL Anywhere 11]。
5. 单击 [下一步]。
6. 请按照 [创建新数据源向导] 中的说明进行操作。

在 Windows Mobile 上使用 ODBC 数据源

Windows Mobile 不提供 ODBC 驱动程序管理器或 [ODBC 管理器]。在 Windows Mobile 上，SQL Anywhere 使用存储在文件中的 ODBC 数据源。您可通过使用 DSN 或 FILEDSN 关键字来使用这些数据源定义；在 Windows Mobile 上，DSN 和 FILEDSN 是同义词。

小心

不建议在数据源中存储用户 ID、加密或未加密口令以及数据库密钥。

数据源位置

Windows Mobile 在设备的根目录中搜索数据源文件：*filename.dsn*。

每一个数据源都保存在一个文件中。该文件与数据源同名，扩展名为 *.dsn*。

另请参见

- [“在 Windows 上使用文件数据源”一节第 94 页](#)

Windows Mobile 数据源示例

以下是 Windows Mobile 的 ODBC 数据源示例。

```
[ODBC]
DRIVER=\windows\dbodbc11.dll
UID=DBA
PWD=sql
Integrated=No
AutoStop=Yes
ServerName=SalesDB_remote
LINKS=tcpip(host=192.168.0.55;port=2638;dobroadcast=none)
LOG=\sa_connection.txt
START=dbsrv11 -c 8M
```

另请参见

- [“创建 ODBC 数据源以连接到 Windows Mobile 设备”一节第 319 页](#)

在 Unix 上使用 ODBC 数据源

在 Unix 操作系统上，ODBC 数据源保存在系统信息文件中。该文件可能（也可能不）命名为 *.odbc.ini*。系统会按顺序在以下位置搜索系统信息文件：

- ODBCINI 环境变量。
- ODBC_INI 环境变量。
- ODBCHOME 环境变量。
- HOME 环境变量。
- 用户的主目录 (~)。
- PATH 环境变量。

注意

ODBCINI 和 ODBC_INI 环境变量均指向系统信息文件（其名称可能是也可能不是 *.odbc.ini*），而 ODBC_HOME 和 HOME 环境变量则指向 *.odbc.ini* 文件所在的路径。

ODBCINI 和 ODBC_INI 都指定完整路径，包括文件名。如果系统信息文件位于由 ODBCINI 或 ODBC_INI 指定的目录中，则不必将其命名为 *.odbc.ini*。

以下是一个示例系统信息文件：

```
[My Data Source]
ENG=myserver
CommLinks=tcpip(Host=hostname)
UID=DBA
PWD=sql
```

您可以在系统信息文件中输入任何连接参数。请参见“[连接参数](#)”一节第 248 页。

网络协议选项作为 CommLinks (LINKS) 参数的一部分添加。请参见“[网络协议选项](#)”一节第 284 页。

小心

不建议在数据源中存储用户 ID、加密或未加密口令以及数据库密钥。

在 Unix 上，可以使用 dbdsn 实用程序创建和管理 ODBC 数据源。

小心

在 Unix 上，除非仅使用 SQL Anywhere 数据源，否则，不应使用文件隐藏实用程序 (dbfhide) 为系统信息文件（缺省名称为 *.odbc.ini*）进行简单加密。如果计划使用其它数据源（例如，为了 MobiLink 同步），那么对系统信息文件内容进行模糊处理可能会妨碍其它驱动程序正常工作。

另请参见

- “[创建 ODBC 数据源](#)”一节第 90 页
- “[数据源实用程序 \(dbdsn\)](#)”一节第 710 页
- “[ODBC_HOME 环境变量 \[Unix\]](#)”一节第 349 页
- “[ODBCINI 和 ODBC_INI 环境变量 \[Unix\]](#)”一节第 350 页

使用 OLE DB 连接到数据库

本节介绍在以下环境中，如何使用 OLE DB 连接 SQL Anywhere 数据库：

- Microsoft ActiveX 数据对象（ActiveX Data Objects，简称 ADO）为 OLE DB 数据源提供了编程接口。您可以通过 Microsoft Visual Basic 之类的编程工具访问 SQL Anywhere。
- Sybase PowerBuilder 可以访问 OLE DB 数据源，您可以将 SQL Anywhere 用作 PowerBuilder OLE DB 数据库配置文件。

OLE DB 使用组件对象模型 (COM) 使应用程序可以使用各种来源的数据。关系数据库是可以通过 OLE DB 访问的其中一类数据源。

另请参见

- “OLE DB 介绍”一节 《SQL Anywhere 服务器 - 编程》

OLE DB 提供程序

您想要访问的每类数据源都需要一个 OLE DB 提供程序。每个 **OLE DB 提供程序** 都是一个动态链接库。要访问 SQL Anywhere，请选择以下 OLE DB 提供程序之一：

- **Sybase SQL Anywhere OLE DB 提供程序** SQL Anywhere OLE DB 提供程序作为 OLE DB 数据源提供对 SQL Anywhere 的访问，而无需使用 ODBC 组件。该提供程序的简称是 **SAOLEDB**。

SAOLEDB 提供程序是自行注册的。此注册过程包括在注册表的 COM 部分创建注册表条目，这样在调用 SAOLEDB 提供程序时 ADO 就可以找到 DLL。如果更改 DLL 的位置，则必须重新注册。

- **ODBC 的 Microsoft OLE DB 提供程序** Microsoft 提供一个简称为 **MSDASQL** 的 OLE DB 提供程序。

MSDASQL 提供程序使 ODBC 数据源以 OLE DB 数据源的形式出现。它需要使用 SQL Anywhere ODBC 驱动程序。

另请参见

- “OLE DB 介绍”一节 《SQL Anywhere 服务器 - 编程》

从 ADO 连接

ADO 是一种面向对象的编程接口。在 ADO 中，**Connection** 对象代表与数据源建立的唯一会话。

您可以使用以下 Connection 对象的特性来启动连接：

- 用于保存提供程序名称的 Provider 属性。如果您不提供 Provider 名称，ADO 将使用 MSDASQL 提供程序。

- 用于保存连接字符串的 `ConnectionString` 属性。该属性保存 SQL Anywhere 连接字符串，其使用方式与 ODBC 驱动程序相同。您可以提供 ODBC 数据源名称，也可以明确指定 `UserID`、`Password`、`DatabaseName` 和其它参数，就像使用其它连接字符串一样。
- `Open` 方法用于启动连接。

示例

以下 Visual Basic 代码启动与 SQL Anywhere 的 OLE DB 连接：

```
' Declare the connection object
Dim myConn as New ADODB.Connection
myConn.Provider = "SAOLEDB"
myConn.ConnectionString = "DSN=SQL Anywhere 11 Demo"
myConn.Open
```

另请参见

- “利用 SQL Anywhere 进行 ADO 编程” 一节 《SQL Anywhere 服务器 - 编程》

使用集成登录

集成登录功能允许操作系统、网络登录和数据库连接使用一个用户 ID 和口令。要创建集成登录：

- 启用集成登录功能。
- 创建一个数据库用户，用来映射集成登录（如果不存在任何用户）。
- 创建 Windows 用户或组配置文件与现有的数据库用户之间的集成登录映射。Sybase Central 中的 [登录映射] 文件夹列出了具有集成登录权限的所有用户。
- 从客户端应用程序连接并测试集成登录功能。

支持的操作系统

基于 Windows 的数据库服务器可以使用集成登录功能。Windows 客户端能够使用集成登录连接到运行于 Windows 上的网络服务器。

集成登录的优点

集成登录是一个或多个 Windows 用户或 Windows 用户组配置文件到现有数据库用户的映射。如果一名用户已经成功通过对其用户（或组）配置文件的安全验证，并且已经登录到计算机上，那么，这名用户不用另外提供用户 ID 或口令即可连接到数据库。

要进行此操作，必须将数据库配置为使用集成登录，且必须在用于登录到计算机或网络的用户（或组）配置文件与数据库用户之间建立一个映射。

使用集成登录对于用户更加方便，且允许数据库和网络安全具有统一的安全系统。集成登录的优点包括：

- 用户不必键入用户 ID 或口令。
- 用户由操作系统来验证。单一系统用于数据库安全和计算机或网络安全。
- 多个用户或组配置文件可映射到单个数据库用户 ID。
- 用于登录到 Windows 计算机的名称和口令不必与数据库用户 ID 和口令一致。

小心

集成登录为统一的安全系统提供了方便，但是它在安全性上涉及到一些值得引起关注的负效应，数据库管理员应当熟悉这些负效应。请参见“安全问题：不受限制的数据库访问”一节第 105 页和“安全问题：复制的数据库文件”一节第 115 页。

启用集成登录功能

`login_mode` 数据库选项确定是否启用集成登录功能。因为数据库选项只应用于它们所在的数据库，所以，不同的数据库即使在同一服务器内装载和运行，也可以有不同的集成登录设置。

`login_mode` 数据库选项接受以下值：

- **Standard** 允许标准登录。这是缺省设置。标准连接登录必须提供用户 ID 和口令，且不使用集成或 Kerberos 连接参数。如果尝试进行集成或 Kerberos 登录连接，则会发生错误。
- **集成** 允许集成登录。
- **Kerberos** 允许 Kerberos 登录。请参见“[Kerberos 验证](#)”一节第 107 页。

小心

将 `login_mode` 数据库选项设置为不允许标准登录可限制只有被授予了集成或 Kerberos 登录映射的用户或组才能进行连接。如果用户不具有 DBA 权限，则尝试使用用户 ID 和口令连接会生成错误。

要允许多种登录类型，请为 `login_mode` 选项指定多个值。例如，以下 SQL 语句用于将 `login_mode` 数据库选项的值设置为同时允许标准登录和集成登录：

```
SET OPTION PUBLIC.login_mode = 'Standard,Integrated';
```

如果可以复制数据库文件，则应使用临时公共 `login_mode` 选项（集成登录和 Kerberos 登录均应使用）。这样，如果文件被复制，缺省情况下将不再支持集成登录和 Kerberos 登录。

创建集成登录

用户配置文件只能映射到现有的数据库用户 ID。当该数据库用户 ID 从数据库中删除时，基于该数据库用户 ID 的所有集成登录映射都将被自动删除。

用户或组配置文件不必仅为映射到某个数据库 ID 而存在。多个用户配置文件可映射到同一个数据库用户 ID。

可使用 [\[创建登录映射向导\]](#) 或 SQL 语句创建集成登录映射。

◆ 映射集成登录 (Sybase Central):

要创建或删除集成登录映射，用户必须具有 DBA 权限。

1. 打开 Sybase Central。
2. 以具有 DBA 权限的用户身份连接到数据库。
3. 在左窗格中，右击 [\[登录映射\]](#) » [\[新建\]](#) » [\[登录映射\]](#)。
4. 单击 [\[下一步\]](#)。
5. 在 [\[哪个 Windows 用户将连接到数据库\]](#) 字段中，键入将要为其创建集成登录的用户或组配置文件的名称。
6. 在 [\[您希望哪个数据库用户与 Windows 用户相关联\]](#) 列表中，选择该用户映射到的数据库用户 ID。
7. 请按照 [\[创建登录映射向导\]](#) 中的其余说明进行操作。

◆ 映射集成登录 (SQL):

1. 以具有 DBA 权限的用户身份连接到数据库。

2. 执行 GRANT INTEGRATED LOGIN TO 语句。

示例

以下 SQL 语句允许 Windows 用户 fran_whitney 和 matthew_cobb 以用户 DBA 身份登录到数据库，而不必知道或提供 DBA 用户 ID 或口令。

```
GRANT INTEGRATED LOGIN
TO fran_whitney, matthew_cobb
AS USER DBA;
```

请参见“GRANT 语句”一节《SQL Anywhere 服务器 - SQL 参考》。

以下 SQL 语句允许属于 Windows NT 组 mywindowsusers 的成员的 Windows 用户以用户 DBA 身份登录到数据库，而不必知道或提供 DBA 用户 ID 或口令。

```
GRANT INTEGRATED LOGIN
TO mywindowsusers
AS USER DBA;
```

请参见“为 Windows 用户组创建集成登录”一节第 102 页。

撤消集成登录权限

◆ 撤消集成登录权限 (Sybase Central):

1. 打开 Sybase Central。
2. 以具有 DBA 权限的用户身份连接到数据库。
3. 在左窗格中，单击 [登录映射]。
4. 在右窗格中，右击您想要删除的登录映射，然后单击 [删除]。
5. 单击 [是]。

◆ 撤消集成登录权限 (SQL):

1. 以具有 DBA 权限的用户身份连接到数据库。
2. 执行 REVOKE INTEGRATED LOGIN FROM 语句。

示例

以下 SQL 语句用于删除 Windows 用户 pchin 的集成登录权限。

```
REVOKE INTEGRATED LOGIN
FROM pchin;
```

请参见“REVOKE 语句”一节《SQL Anywhere 服务器 - SQL 参考》。

从客户端应用程序连接到数据库

要使用集成登录将客户端应用程序连接到数据库：

- 将连接参数列表中的 `Integrated (INT)` 参数设置为 `YES`。
- 在连接字符串或 **[连接]** 窗口中既不指定用户 ID，也不指定口令。

如果将连接字符串中的 `Integrated (INT)` 参数设置为 `YES`，会尝试进行集成登录。当连接尝试失败且 `login_mode` 数据库选项设置为 `[Standard, Integrated]` 时，服务器会尝试进行标准登录。请参见“[login_mode 选项 \[数据库\]](#)”一节第 512 页。

如果试图在不提供用户 ID 或口令的情况下连接到数据库，则会尝试进行集成登录。登录尝试成功与否取决于当前用户配置文件名是否与数据库中的集成登录映射相匹配。

Interactive SQL 示例

在下例中，当用户登录时所使用的配置文件与缺省数据库服务器中的集成登录映射相匹配时，连接尝试便取得了成功：

```
CONNECT USING 'INTEGRATED=yes';
```

当存在以下情况时，可使用 Interactive SQL 语句 `CONNECT` 连接到数据库：

- 服务器当前正在运行。
- 缺省数据库已将 `login_mode` 数据库选项设置为接受集成登录连接。
- 已创建与当前用户的用户配置文件名匹配的集成登录映射，或已为该用户所属的 Windows 用户组创建了集成登录映射。
- 当出现更多的连接信息提示时，用户只需单击 **[确定]** 而不必提供更多的信息。

为 Windows 用户组创建集成登录

当 Windows 用户登录时，如果他们没有显式集成登录映射，但属于一个有集成登录映射的 Windows 用户组，则该用户将作为 Windows 用户组的集成登录映射中指定的数据库用户或组连接到该数据库。

小心

创建某个 Windows 用户组的集成登录后，所有该组成员用户无需提供用户 ID 或口令就能连接到该数据库。

请参见“[防止 Windows 用户组成员连接到数据库](#)”一节第 103 页。

多个组的成员

如果 Windows 用户属于多个 Windows 用户组，且计算机上的这些多个 Windows 用户组在数据库中具有集成登录映射，则只有在计算机上的所有这些 Windows 用户组具有到同一数据库用户 ID 的集成登录映射时，集成登录才能成功。如果这些多个 Windows 用户组具有到不同数据库用户 ID 的集成登录映射，则会返回一个错误，并且集成登录会失败。

例如，请看这样一个数据库：它有两个用户 ID（dbuserA 和 dbuserB）和一个 Windows 用户 windowsuser，该用户属于 Windows 用户组 xpgroupA 和 xpgroupB。

此 SQL 语句...	允许...
<pre>GRANT INTEGRATED LOGIN TO windowsuser AS USER dbuserA;</pre>	windowsuser 使用为 windowsuser 显式设置的集成登录映射连接到该数据库。
<pre>GRANT INTEGRATED LOGIN TO xpgroupA AS USER dbuserB;</pre>	windowsuser 使用授予 xpgroupA 的集成登录映射连接到该数据库。
<pre>GRANT INTEGRATED LOGIN TO xpgroupA AS USER dbuserB; GRANT INTEGRATED LOGIN xpgroupB AS USER dbuserB;</pre>	windowsuser 连接到该数据库，因为 windowsuser 所属的所有 Windows 用户组都具有到相同数据库用户的集成登录映射。
<pre>GRANT INTEGRATED LOGIN TO xpgroupA AS USER dbuserA; GRANT INTEGRATED LOGIN TO xpgroupB AS USER dbuserB;</pre>	与数据库之间无连接。当 windowsuser 尝试连接到该数据库时，集成登录失败，因为两个 Windows 用户组具有到不同数据库用户的集成登录映射，并且 windowsuser 是这两个 Windows 用户组的成员。

域控制器位置

缺省情况下，运行 SQL Anywhere 数据库服务器的计算机用于验证 Windows 用户组成员资格。如果域控制器服务器不在运行数据库服务器的计算机上，则可以使用 `integrated_server_name` 选项来指定域控制器服务器的名称。例如：

```
SET OPTION PUBLIC.integrated_server_name = '\\myserver-1';
```

请参见“[integrated_server_name 选项 \[数据库\]](#)”一节第 507 页。

防止 Windows 用户组成员连接到数据库

有两种方法可以防止具有集成登录的 Windows 用户组中的成员使用该组的集成登录连接到数据库：

- 为该用户创建一个到没有口令的数据库用户 ID 的集成登录。
- 创建一个由 `login_procedure` 选项调用的存储过程，用于检查是否允许用户登录，并在被禁止的用户试图连接时抛出异常。

创建到没有口令的用户 ID 的集成登录

如果用户是一个具有集成登录的 Windows 用户组的成员，同时其用户 ID 具有显式集成登录，则使用该用户的集成登录连接到数据库。要防止用户使用他们的 Windows 用户组集成登录连接到数据库，您可以为该 Windows 用户创建一个到没有口令的数据库用户 ID 的集成登录。没有口令的数据库用户 ID 不能连接到数据库。

◆ 创建到没有口令的用户 ID 的集成登录

1. 将用户添加到数据库，但不提供口令。例如：

```
CREATE USER db_user_no_password;
```

2. 为该 Windows 用户创建映射到没有口令的数据库用户的集成登录。例如：

```
GRANT INTEGRATED LOGIN TO WindowsUser  
AS USER db_user_no_password;
```

创建过程来防止 Windows 用户进行连接

login_procedure 选项指定每次尝试连接数据库时要调用的存储过程。缺省情况下，调用 dbo.sp_login_environment 过程。可将 login_procedure 选项设置为调用为防止特定用户连接数据库而编写的过程。

以下示例创建一个名为 login_check 的过程，供 login_procedure 选项调用。login_check 过程在禁止连接到数据库的用户的列表中检查提供的用户名。如果在列表中找到了提供的用户名，则连接失败。在此示例中，名为 Joe、Harry 或 Martha 的用户被禁止连接。如果未在列表中找到该用户，则数据库连接照常进行，并调用 sp_login_environment 过程。

```
CREATE PROCEDURE DBA.user_login_check()  
BEGIN  
    DECLARE INVALID_LOGON EXCEPTION FOR SQLSTATE '28000';  
    // Disallow certain users  
    IF( CURRENT USER IN ('Joe','Harry','Martha') ) THEN  
        SIGNAL INVALID_LOGON;  
    ELSE  
        CALL sp_login_environment;  
    END IF;  
END  
go  
GRANT EXECUTE ON DBA.user_login_check TO PUBLIC  
go  
SET OPTION PUBLIC.login_procedure='DBA.user_login_check'  
go
```

关于网络上的集成登录

如果该数据库位于网络服务器上，则要使用的集成登录必须满足下列两个条件之一：

- 用于集成登录连接尝试的用户配置文件在本地计算机和服务器上必须是相同的。本地计算机和服务器上的用户配置文件的口令也必须相同。

例如，当用户 jsmith 试图使用集成登录连接到网络服务器上装载的数据库时，则本地计算机和运行数据库服务器的计算机必须具有相同的用户配置文件名和口令。必须允许用户 jsmith 登录到两台计算机。

- 如果网络访问由 Microsoft 域来控制，试图进行集成登录的用户必须对域控制器服务器拥有域权限且必须登录到网络中。网络服务器上的用户配置文件无需与本地计算机上的用户配置文件相匹配。

创建缺省的集成登录用户

您可以创建缺省的集成登录用户 ID，这样，即使对当前正在使用的用户配置文件不存在集成登录映射，也可以通过集成登录成功连接。

例如，如果对用户配置文件名 JSMITH 不存在集成登录映射，则当 JSMITH 是正在使用的用户配置文件时，集成登录连接尝试通常会失败。

但是，如果您在数据库中创建一个名为 Guest 的用户 ID，则即使没有显式地为用户配置文件 JSMITH 建立集成登录映射，集成登录将成功映射到 Guest 用户 ID。

小心

缺省集成登录用户允许任何尝试集成登录的用户成功连接到包含名为 Guest 的用户 ID 的数据库。授予 Guest 用户 ID 的权限决定了授予新连接的用户的权限和特权。

安全问题：不受限制的数据库访问

集成登录功能使用 Windows 的登录控制系统代替 SQL Anywhere 安全系统来工作，使得无需提供用户 ID 或口令便可连接数据库。实质上，如果用户可以登录到承载数据库的计算机，他就能够通过数据库安全验证。

如果用户以 dsmith 身份成功登录到 Windows 服务器，只要用户有集成登录映射或者缺省的集成登录用户 ID，则他们不必经过进一步的身份验证即可连接数据库。

在使用集成登录时，数据库管理员应当特别注意 Windows 强制登录安全的方法，以防止出现对数据库不必要的访问。

小心

如果启用 Guest 用户配置文件，将允许对在该服务器上运行的数据库进行不受限制的访问。

如果 Guest 用户配置文件处于启用状态且口令为空，则登录到该服务器的任何尝试都将成功。用户配置文件无需存在于服务器上，或者所提供的登录 ID 无需具有域登录权限。准确地说，任何用户都可以使用任何登录 ID 和任何口令登录到服务器：缺省情况下他们会登录到 Guest 用户配置文件。

这对于在启用集成登录功能的情况下连接到数据库有着重要的含义。

请考虑以下情况：假定承载数据库的 Windows 服务器有一个用空口令启用的 Guest 用户配置文件。

- 在用户 fran_whitney 和数据库用户 ID DBA 之间存在集成登录映射。用户 fran_whitney 使用正确的登录 ID 和口令连接服务器时，她将以 DBA 身份（具有全部管理权限的用户）连接到数据库。

但是，任何试图以 fran_whitney 身份连接服务器的人都将成功登录到服务器，而无论他们提供什么口令，这是因为 Windows 将该连接尝试缺省为 Guest 用户配置文件。在使用 fran_whitney 登录 ID 成功登录到服务器之后，未经授权的用户便可以通过集成登录映射，以 DBA 身份成功连接到数据库。

为了安全而禁用 Guest 用户配置文件

最安全的集成登录策略是在承载 SQL Anywhere 数据库的所有 Windows 计算机上禁用 Guest 用户配置文件。这可以通过使用 Windows 用户管理器实用程序来实现。

Kerberos 验证

Kerberos 登录功能允许您对数据库连接、操作系统和网络登录维护单个用户 ID 和口令。Kerberos 登录使用户更加方便，且允许数据库和网络安全体系具有统一的安全系统。它的优点包括：

- 用户连接数据库时无需提供用户 ID 或口令。
- 多个用户可映射到单个数据库用户 ID。
- 用于登录到 Kerberos 的名称和口令不必与数据库用户 ID 和口令一致。

Kerberos 是一种网络验证协议，它使用密钥加密算法提供强验证和强加密。已经登录到 Kerberos 的用户无需提供用户 ID 或口令即可连接到数据库。

可使用 Kerberos 进行验证。要将验证委派给 Kerberos，您必须：

- 将服务器和数据库配置为使用 Kerberos 登录
- 创建登录到计算机或网络的用户 ID 与数据库用户之间的映射

小心

将 Kerberos 登录用作单独的安全解决方案时，需要考虑一些值得引起关注的安全性方面的负效应。请参见“[安全问题：复制的数据库文件](#)”一节第 115 页。

SQL Anywhere 中不包含 Kerberos 软件；该软件必须单独获取。Kerberos 软件中包含以下组件：

- **Kerberos 库** 这些被称作 Kerberos 客户端或 GSS（通用安全服务）-API 运行时库。这些 Kerberos 库可执行明确定义的 GSS-API。每个想要使用 Kerberos 的客户端和服务端计算机上都必须有这些库。如果您使用 Active Directory 作为您的 KDC，则可以使用内置的 Windows SSPI 接口来取代第三方 Kerberos 客户端库。
- **Kerberos 密钥分发中心 (KDC) 服务器** KDC 起着用户和服务器的仓库的作用。同时它还能验证用户和服务器的身份标识。KDC 通常安装在在不应用程序或用户登录的服务器计算机上。

SQL Anywhere 支持通过 DBLib、ODBC、OLE DB 和 ADO.NET 客户端以及 Sybase Open Client 和 jConnect 客户端进行 Kerberos 验证。Kerberos 验证可以与 SQL Anywhere 传送层安全加密配合使用，但 SQL Anywhere 不支持网络通信的 Kerberos 加密。

Windows 使用 Kerberos 作为 Windows 域和域帐户。Active Directory Windows 域控制器执行 Kerberos KDC。数据库服务器计算机仍然需要第三方 Kerberos 客户端或运行时库才能在此环境下进行验证，但 Windows 客户端计算机可使用内置的 Windows SSPI 接口来取代第三方 Kerberos 客户端或运行时库。请参见“[在 Windows 上使用 SSPI 进行 Kerberos 登录](#)”一节第 112 页。

Kerberos 客户端

可以在 32 位 Windows 和 Linux 系统上进行 Kerberos 验证。有关经过测试的 Kerberos 客户端的列表，请参见 <http://www.sybase.com/detail?id=1062626>。

下表列出了所支持的 Kerberos 客户端使用的 keytab 和 GSS-API 文件的缺省名称和位置。

Kerberos 客户端	缺省 keytab 文件	GSS-API 库文件名	注意
Windows MIT Kerberos 客户端	<i>C:\WINDOWS\krb5kt</i>	<i>gssapi32.dll</i>	在启动数据库服务器前，可以设置 KRB5_KTNAME 环境变量，以指定不同的 keytab 文件。
Windows CyberSafe Kerberos 客户端	<i>C:\Program Files\CyberSafe v5srvtab</i>	<i>gssapi32.dll</i>	在启动数据库服务器前，可以设置 CSFC5KTNAME 环境变量，以指定不同的 keytab 文件。
Unix MIT Kerberos 客户端	<i>/etc/krb5.keytab</i>	<i>libgssapi_krb5.so</i> ¹	在启动数据库服务器前，可以设置 KRB5_KTNAME 环境变量，以指定不同的 keytab 文件。
Unix CyberSafe Kerberos 客户端	<i>/krb5/v5srvtab</i>	<i>libgss.so</i> ¹	在启动数据库服务器前，可以设置 CSFC5KTNAME 环境变量，以指定不同的 keytab 文件。
Unix Heimdal Kerberos 客户端	<i>/etc/krb5.keytab</i>	<i>libgssapi.so.1</i> ¹	

¹ 这些文件名可能会有所不同，这取决于您的操作系统和 Kerberos 客户端版本。

设置 Kerberos 验证

◆ 在 SQL Anywhere 数据库上设置 Kerberos 验证

- 在客户端和服务上都安装和配置 Kerberos 客户端软件，包括 GSS-API 运行时库。
 在使用 Active Directory KDC 的 Windows 客户端计算机上，您可以使用 SSPI，不必安装 Kerberos 客户端。请参见“在 Windows 上使用 SSPI 进行 Kerberos 登录”一节第 112 页。
- 如有必要，请在 Kerberos 密钥分发中心（Key Distribution Center，简称 KDC）内为每位用户创建一个 Kerberos 主体。
 Kerberos 主体就是一个 Kerberos 用户 ID，其格式为 *user/instance@REALM*，其中 */instance* 为可选项。如果您正在使用 Kerberos，则主体应该已经存在，因此不必为每位用户创建一个 Kerberos 主体。
 主体区分大小写，因此必须指定正确的大小写。不支持映射只有大小写存在差异的多个主体（例如，无法同时映射 *jjordan@MYREALM.COM* 和 *JJordan@MYREALM.COM*）。
- 在 SQL Anywhere 数据库服务器的 KDC 内创建 Kerberos 主体。

数据库服务器 Kerberos 主体的格式为 *server-name@REALM*，其中 *server-name* 指 SQL Anywhere 数据库服务器的名称。主体区分大小写，并且 *server-name* 不能包含多字节字符或者 /、\ 或 @。在其余步骤中，假设 Kerberos 主体为 *my_server_princ@MYREALM.COM*。

因为服务器使用 *keytab* 文件进行 KDC 验证，所以必须在 KDC 内创建服务器服务主体。*keytab* 文件是受保护和已加密的。

4. 将主体 *server-name@REALM* 的 *keytab* 从 KDC 安全地提取并复制到运行 SQL Anywhere 数据库服务器的计算机上。*Keytab* 文件的缺省位置取决于 Kerberos 客户端和平台。*keytab* 文件的权限应设置为 SQL Anywhere 服务器能够读取该文件，而其他未授权的用户则没有读取权限。
5. 将 SQL Anywhere 配置为使用 Kerberos

将 SQL Anywhere 配置为使用 Kerberos

1. 在 SQL Anywhere 数据库上设置 Kerberos 验证。请参见“设置 Kerberos 验证”一节第 108 页。
2. 使用 *-krb* 或 *-kr* 选项启动 SQL Anywhere 服务器，以启用 Kerberos 验证，或者使用 *-kl* 选项指定 GSS-API 库的位置并启用 Kerberos。
3. 将公共或临时公共选项 *login_mode* 更改为包含 Kerberos 的值。您必须具有 DBA 权限才能更改此选项的设置。*Login_mode* 数据库选项确定是否允许 Kerberos 登录。因为数据库选项只作用于它们所在的数据库，所以，不同的数据库即使是在同一服务器内装载和运行，也可以有不同的 Kerberos 登录设置。例如：

```
SET OPTION PUBLIC.login_mode = 'Kerberos,Standard';
```

login_mode 数据库选项接受一个或多个下列值：

- **Standard** 允许标准登录。这是缺省值。标准连接登录必须提供用户 ID 和口令，且不使用集成或 Kerberos 连接参数。
- **Integrated** 允许集成登录。
- **Kerberos** 允许 Kerberos 登录。

小心

将 *login_mode* 数据库选项设置为 Kerberos，会使只有被授予了 Kerberos 登录映射的用户才能进行连接。如果用户不具有 DBA 权限，则尝试使用用户 ID 和口令连接会生成错误。

4. 为客户端创建数据库用户 ID。只要现有数据库用户拥有正确的权限，您就可以使用该用户的 ID 进行 Kerberos 登录。例如：

```
CREATE USER "kerberos-user"  
IDENTIFIED BY abc123;
```

5. 执行 GRANT KERBEROS LOGIN TO 语句，以创建从客户端的 Kerberos 主体到现有数据库用户 ID 的映射。此语句需要 DBA 权限。例如：

```
GRANT KERBEROS LOGIN TO "pchin@MYREALM.COM"  
AS USER "kerberos-user";
```

如果要在使用的 Kerberos 主体没有映射的情况下进行连接，请确保具有 Guest 数据库用户 ID 且拥有口令。请参见“[创建缺省的集成登录用户](#)”一节第 105 页。

6. 确保客户端用户已经使用其 Kerberos 主体登录（具有有效的 Kerberos 票据授予票据）且客户端的 Kerberos 票据未过期。Windows 用户登录到已拥有票据授予票据的域帐户，使他们可以对服务器进行验证（假设他们的主体拥有足够权限）。

票据授予票据是一种通过用户口令加密的 Kerberos 票据，票据授予服务使用该票据来验证用户的身份。

7. 从客户端连接，指定 KERBEROS 连接参数（通常为 KERBEROS=YES，但也可以使用 KERBEROS=SSPI 或 KERBEROS=GSS-API-library-file）。如果指定了用户 ID 或口令连接参数，则它们将被忽略。例如：

```
dbisql -c "KERBEROS=YES;ENG=my_server_princ"
```

Interactive SQL 示例

例如，某个用户已经登录，且登录时使用的用户配置文件名与服务器的缺省数据库中的 Kerberos 登录映射相匹配，那么，使用以下 Interactive SQL 语句尝试连接就会成功：

```
CONNECT USING 'KERBEROS=YES';
```

满足下列所有条件时，Interactive SQL 语句 CONNECT 即可以连接到数据库：

- 服务器当前正在运行。
- 当前服务器上的缺省数据库能够接受使用 Kerberos 验证的连接。
- 已经为用户的当前 Kerberos 主体创建 Kerberos 登录映射。
- 如果服务器显示一个窗口，提示用户输入更多连接信息（例如，使用 Interactive SQL 时出现这种情况），用户单击 [确定] 并且不提供更多信息。

另请参见

- “[-kl 服务器选项](#)”一节第 190 页
- “[-kr 服务器选项](#)”一节第 191 页
- “[-krb 服务器选项](#)”一节第 192 页
- “[login_mode 选项 \[数据库\]](#)”一节第 512 页
- “[GRANT 语句](#)”一节《[SQL Anywhere 服务器 - SQL 参考](#)》
- “[创建 Kerberos 登录映射](#)”一节第 111 页

从 Open Client 或 jConnect 应用程序连接

要从 Open Client 或 jConnect 应用程序连接：

- 设置 Kerberos 验证。请参见“[设置 Kerberos 验证](#)”一节第 108 页。
- 将 SQL Anywhere 配置为使用 Kerberos。请参见“[将 SQL Anywhere 配置为使用 Kerberos](#)”一节第 109 页。

- 使用 Adaptive Server Enterprise 为 Kerberos 验证设置 Open Client 或 jConnect（按照您期望的方式）。服务器名称必须是 SQL Anywhere 服务器的名称且区分大小写。不能使用备用服务器名称从 Open Client 或 jConnect 连接。

有关设置 Kerberos 主体和提取 keytab 的信息，请参见 <http://www.sybase.com/detail?id=1029260>。

另请参见

- “-krb 服务器选项”一节第 192 页
- “-kr 服务器选项”一节第 191 页
- “-kl 服务器选项”一节第 190 页
- “login_mode 选项 [数据库]”一节第 512 页
- “GRANT 语句”一节 《SQL Anywhere 服务器 - SQL 参考》
- “CREATE USER 语句”一节 《SQL Anywhere 服务器 - SQL 参考》
- “Kerberos 连接参数 [KRB]”一节第 270 页
- “Kerberos 连接疑难解答”一节第 113 页

创建 Kerberos 登录映射

◆ 创建 Kerberos 登录映射 (Sybase Central)

1. 打开 Sybase Central。
2. 以具有 DBA 权限的用户身份连接到数据库。
3. 在左窗格中，右击 [登录映射] » [新建] » [登录映射]。
4. 请按照 [创建登录映射向导] 中的说明进行操作。

◆ 创建 Kerberos 登录映射 (SQL)

1. 以具有 DBA 权限的用户身份连接到数据库。
2. 执行 GRANT KERBEROS LOGIN TO 语句。

请参见“GRANT 语句”一节 《SQL Anywhere 服务器 - SQL 参考》。

示例

以下 SQL 语句可将 KERBEROS 登录权限授予 Windows 用户 pchin。

```
GRANT KERBEROS LOGIN TO "pchin@MYREALM.COM"  
AS USER "kerberos-user";
```

撤消 KERBEROS 登录权限

◆ 撤消 Kerberos 登录映射 (Sybase Central)

1. 打开 Sybase Central。

2. 以具有 DBA 权限的用户身份连接到数据库。
3. 在左窗格中，单击 [登录映射]。
4. 在右窗格中，右击 [登录映射]，然后选择 [删除]。
5. 单击 [是]。

◆ 撤消 Kerberos 登录映射 (SQL)

1. 以具有 DBA 权限的用户身份连接到数据库。
2. 执行 REVOKE KERBEROS LOGIN FROM 语句。
请参见“REVOKE 语句”一节《SQL Anywhere 服务器 - SQL 参考》。

示例

以下 SQL 语句删除 Windows 用户 pchin 的 KERBEROS 登录权限。

```
REVOKE KERBEROS LOGIN  
FROM "pchin@MYREALM.COM";
```

在 Windows 上使用 SSPI 进行 Kerberos 登录

在 Windows 域中，可以在基于 Windows 的计算机上使用 SSPI 且无需在客户端计算机上安装 Kerberos 客户端。Windows 域帐户已经拥有关联的 Kerberos 主体。

◆ 使用 SSPI 连接

1. 设置 Kerberos 验证。请参见“设置 Kerberos 验证”一节第 108 页。
2. 使用 -krb 选项启动 SQL Anywhere 服务器以便启用 Kerberos 验证。例如：

```
dbeng11 -krb -n my_server_princ C:\kerberos.db
```
3. 将公共或临时公共选项 login_mode 更改为包含 Kerberos 的值。必须具有 DBA 权限才能设置此选项。例如：

```
SET OPTION PUBLIC.login_mode = 'Kerberos';
```

4. 为客户端创建数据库用户 ID。只要现有数据库用户拥有正确的权限，您就可以使用该用户的 ID 进行 Kerberos 登录。例如：

```
CREATE USER kerberos_user  
IDENTIFIED BY abc123;
```

5. 执行 GRANT KERBEROS LOGIN TO 语句，创建从客户端的 Kerberos 主体到现有数据库用户 ID 的映射。此语句需要 DBA 权限。例如：

```
GRANT KERBEROS LOGIN TO "pchin@MYREALM.COM"  
AS USER "kerberos-user";
```

6. 从客户端计算机连接数据库。例如：

```
dbisql -c "KERBEROS=SSPI;ENG=my_server_princ"
```

若连接字符串中指定了 Kerberos=SSPI，则会尝试进行 Kerberos 登录。

某个用户已经登录，且登录时使用的用户配置文件名与服务器的缺省数据库中的 Kerberos 登录映射相匹配，那么，使用以下 Interactive SQL 语句来尝试连接同样也会成功：

```
CONNECT USING 'KERBEROS=SSPI';
```

Kerberos 连接疑难解答

如果尝试启用或使用 Kerberos 验证时出现意外错误，建议在数据库服务器和客户端上均启用附加诊断消息。

如果在启动数据库服务器时指定 -z 选项，或者在服务器已经运行时执行 CALL sa_server_option('DebuggingInformation', 'ON')，数据库服务器消息日志中将包含附加诊断消息。LogFile 连接参数可使客户端诊断信息写入指定的文件。除了使用 LogFile 连接参数之外，也可以执行命令 dbping -z。-z 参数显示有助于标识连接问题根源的诊断消息。

无法启动数据库服务器

症状	常用解决办法
[无法装载 Kerberos GSS-API 库] 消息	<ul style="list-style-type: none"> ● 确保数据库服务器计算机上已经安装 Kerberos 客户端，包括 GSS-API 库 ● 数据库服务器 -z 输出显示正在尝试装载的库名。验证库名是否正确。如有必要，可使用 -kl 选项指定正确的库名。 ● 确保目录和所有支持库都列在库路径（在 Windows 上为 %PATH%）中。 ● 如果数据库服务器 -z 输出显示 GSS-API 库缺少入口点，则该库不是受支持的第 5 版本 Kerberos GSS-API 库。
[无法获取服务器名 "server-name" 的 Kerberos 证书] 消息	<ul style="list-style-type: none"> ● 确保 KDC 中具有 <i>server-name@REALM</i> 的主体。主体区分大小写，因此请确保数据库服务器名与主体名的用户部分大小写一致。 ● 确保 SQL Anywhere 服务器的名称是主体的主要/用户部分。 ● 确保服务器的主体已被提取到一个 keytab 文件中，并且此 keytab 文件位于 Kerberos 客户端的正确位置。请参见“Kerberos 客户端”一节第 107 页。 ● 如果数据库服务器计算机上的 Kerberos 客户端的缺省域与服务器主体内的域不同，请使用 -kr 选项指定服务器主体内的域。
[Kerberos 登录失败] 客户端错误	<ul style="list-style-type: none"> ● 检查数据库服务器的诊断消息。服务器所用的 keytab 文件的某些问题在客户端尝试进行验证时才会发现。

Kerberos 客户端连接疑难解答

如果客户端在尝试使用 Kerberos 验证进行连接时出现错误：

症状	常用解决办法
[不支持 Kerberos 登录] 错误, 并且 LogFile 内包含消息 [未能装载库 Kerberos GSS-API]	<ul style="list-style-type: none"> ● 确保客户端计算机上已安装了 Kerberos 客户端, 包括 GSS-API 库。 ● LogFile 所指定的文件列出尝试装载的库的名称。检查库名是否正确, 如有必要可使用 Kerberos 连接参数指定正确的库名。 ● 确保库路径 (在 Windows 上为 %PATH%) 中包含所有支持库的目录。 ● 如果 LogFile 输出显示 GSS-API 库缺少入口点, 则该库不是受支持的第 5 版本 Kerberos GSS-API 库。
[不支持 Kerberos 登录] 错误	<ul style="list-style-type: none"> ● 确保数据库服务器已经通过指定一个或多个 -krb、-kl 或 -kr 服务器选项启用了 Kerberos 登录。 ● 确保客户端和服务器平台上的 SQL Anywhere 均支持 Kerberos 登录。
[Kerberos 登录失败] 错误	<ul style="list-style-type: none"> ● 确保用户已经登录到 Kerberos 且拥有尚未过期的有效票据授予票据。 ● 确保将客户端计算机和服务器计算机的时间同步到相差不到 5 分钟。
[login_mode 设置不容许登录模式 'Kerberos'] 错误	<ul style="list-style-type: none"> ● login_mode 选项的公共或临时公共数据库选项设置中必须包含值 Kerberos 以允许 Kerberos 登录。
[登录 ID ' <i>client-Kerberos-principal</i> ' 未被映射到任何数据库用户 ID]	<ul style="list-style-type: none"> ● 必须使用 GRANT KERBEROS LOGIN 语句将 Kerberos 主体映射到一个数据库用户 ID。注意, 必须将包括域在内的完整客户端主体提供给 GRANT KERBEROS LOGIN 语句, 只在实例或域内有差异的主体将被区分对待。 ● 或者, 如果希望所有尚未明确映射的有效 Kerberos 主体能建立连接, 可使用 GRANT CONNECT 创建一个带密码的 Guest 数据库用户 ID。

安全问题：为增加安全性而设置临时公共选项

如果使用 SET OPTION 语句为指定数据库设置 login_mode 选项的值以允许标准、集成和 Kerberos 登录, 该语句将为该数据库永久启用指定类型的登录。例如, 以下语句用于永久启用标准登录和集成登录:

```
SET OPTION PUBLIC.login_mode = 'Standard,Integrated';
```

如果关闭并重新启动该数据库，则该选项的值将保持不变，集成登录仍处于启用状态。

使用 SET TEMPORARY OPTION 设置 login_mode 选项仍然允许用户通过集成登录进行访问，但只在数据库未关闭前有效。以下语句可临时更改该选项的值：

```
SET TEMPORARY OPTION PUBLIC.login_mode = 'Standard,Integrated';
```

如果永久选项值为 [标准]，数据库将在关闭时恢复为该值。

设置临时公共选项可为数据库提供附加的安全性。将集成登录或 Kerberos 登录添加到数据库后，数据库将依赖于其运行时所在的操作系统的安全性。如果数据库被复制到另一台计算机，则对于数据库的访问将恢复到 SQL Anywhere 安全模型。

另请参见

- “安全问题：复制的数据库文件”一节第 115 页
- “SET OPTION 语句”一节《SQL Anywhere 服务器 - SQL 参考》

安全问题：复制的数据库文件

如果可以复制数据库文件，则应为集成登录和 Kerberos 登录使用临时公共 login_mode 选项。如果文件被复制，缺省情况下将不再支持集成登录和 Kerberos 登录。

如果数据库内包含敏感信息，则应防止存储数据库文件的计算机受到未经授权的访问。否则，数据库文件可能会被复制，并且可以在其它计算机上对数据进行未经授权的访问。要增加数据库的安全性：

- 使用用户口令（尤其是具有 DBA 权限的口令）复杂难猜。
- 将 PUBLIC.login_mode 数据库选项设置为 Standard。要启用集成登录或 Kerberos 登录，每次启动服务器时应仅更改临时公共选项。这可确保在数据库被复制时只允许进行标准登录。请参见“安全问题：为增加安全性而设置临时公共选项”一节第 114 页。
- 应使用 AES 加密算法强加密数据库文件。加密密钥应复杂难猜。

示例 SQL Anywhere 数据库连接

以下示例说明如何使用 SQL Anywhere 中附带的工具连接到 SQL Anywhere 数据库。

通过 Sybase Central 或 Interactive SQL 连接到示例数据库

◆ 连接到示例数据库 (Sybase Central):

1. 选择 [开始] » [程序] » [SQL Anywhere 11] » [Sybase Central]。
2. 单击 [连接] » [使用 SQL Anywhere 11 连接]。
3. 单击 [ODBC 数据源名称]，然后单击 [浏览]。
4. 选择 [SQL Anywhere 11 Demo]，然后单击 [确定]。

◆ 连接到示例数据库 (Interactive SQL)

1. 选择 [开始] » [程序] » [SQL Anywhere 11] » [Interactive SQL]。
2. 单击 [ODBC 数据源名称]，然后单击 [浏览]。
3. 选择 [SQL Anywhere 11 Demo]，然后单击 [确定]。

注意

您无需为上述连接输入用户 ID 和口令，因为数据源中已经包含这些信息。

◆ 连接到示例数据库（指定数据库文件的位置）

1. 在 Sybase Central 或 Interactive SQL 中，打开 [连接] 窗口。
2. 单击 [标识] 选项卡。
3. 在 [用户 ID] 字段键入 DBA。
4. 在 [口令] 字段中键入 sql。
5. 单击 [数据库] 选项卡。
6. 在 [数据库名] 字段中键入 demo.db。
7. 在 [数据库文件] 字段中，浏览至 *samples-dir*。在 Microsoft Windows XP 操作系统上的缺省位置是 *C:\Documents and Settings\All Users\Shared Documents\SQL Anywhere 11\Samples\demo.db*。
有关 *samples-dir* 的信息，请参见“[示例目录](#)”一节第 368 页。
8. 单击 [确定]。

在 Mac OS X 上连接到示例数据库

Mac OS X 上包括快捷方式。

◆ 从 Interactive SQL 连接到示例数据库 (Mac OS X)

1. 使用 Finder 找到 SQL Anywhere 示例数据库。缺省情况下，它位于 */Applications/SQLAnywhere11/System/demo.db*。
2. 将此文件复制到您具有读写访问权限的位置，如桌面。
3. 在 Finder 中双击 **DBLauncher**。
缺省情况下，**DBLauncher** 位于以下路径中：*/Applications/SQLAnywhere11*。
4. 选择 **[Local Server]**。
[Local Server] 选项不允许通过网络进行客户端/服务器通信。
5. 单击 **[Start]** 启动名为 **demo** 的个人数据库服务器。
6. 在 Finder 中，双击 */Applications/SQLAnywhere11* 中的 **[Interactive SQL]**。
7. 在 **[用户 ID]** 字段键入 **DBA**。
8. 在 **[口令]** 字段中键入 **sql**。
9. 单击 **[确定]**。

连接到本地数据库

使用以下过程之一连接到驻留在您计算机上的数据库。如果已在服务器上装载（启动）数据库，则只需提供数据库名称即可连接到该数据库。不需要指定数据库文件。

要简化数据库访问，请使用连接配置文件。请参见“[Sybase Central 连接配置文件](#)”一节第 88 页。

◆ 连接到本地服务器上正在运行的数据库

1. 启动 Sybase Central 或 Interactive SQL。
如果未出现 **[连接]** 窗口：
 - 在 Sybase Central 中，选择 **[连接]** » **[使用 SQL Anywhere 11 连接]**。
 - 在 Interactive SQL 中，选择 **[SQL]** » **[连接]**。
2. 单击 **[标识]** 选项卡。
3. 在 **[用户 ID]** 字段中键入用户名。
4. 在 **[口令]** 字段中，键入数据库的口令。
5. 如果该服务器正在运行一个数据库，请单击 **[确定]**。
如果该服务器运行多个数据库：

- 单击 [数据库] 选项卡。
- 在 [数据库名] 字段中，键入数据库的名称。
- 单击 [确定]。

◆ 启动并连接到数据库

1. 启动 Sybase Central 或 Interactive SQL。

如果未出现 [连接] 窗口：

- 在 Sybase Central 中，选择 [连接] » [使用 SQL Anywhere 11 连接]。
- 在 Interactive SQL 中，选择 [SQL] » [连接]。

2. 单击 [标识] 选项卡。
3. 在 [用户 ID] 字段中键入用户名。
4. 在 [口令] 字段中，键入数据库的口令。
5. 单击 [数据库] 选项卡。
6. 在 [数据库文件] 字段中，指定文件路径、文件名和文件扩展名，或单击 [浏览] 以浏览数据库文件。
7. 要为后续连接创建一个不同于该文件名的数据库名，请在 [数据库名] 字段中键入一个名称。不要指定文件路径或扩展名。
8. 单击 [确定]。

连接到嵌入式数据库

嵌入式数据库专用于单个应用程序，它在该应用程序所在的计算机上运行，而且通常对用户是隐藏的。

如果应用程序使用嵌入式数据库，则在应用程序进行连接时个人服务器通常不在运行。可以使用连接字符串启动数据库，方法是在连接字符串的 DatabaseFile (DBF) 参数中指定数据库文件。

为了提高自动启动数据库的查询性能，请尽快启动数据库，即使用户不会立即与数据库建立连接。这样，可在对数据库执行查询前，使高速缓存预热。请参见“使用高速缓存预热”一节《SQL Anywhere 服务器 - SQL 的用法》。

使用 DBF 连接参数

DBF 连接参数用于指定所要使用的数据库文件。数据库文件会自动装载到缺省服务器上；或者，如果没有运行的服务器则会启动一个服务器。

不再连接到该数据库时（通常是启动连接的应用程序断开连接时），将会卸载该数据库。如果服务器是由该连接启动的，则该数据库服务器会在数据库卸载之后停止。

在以下示例中，示例数据库将作为嵌入式数据库进行装载：


```
DBF=samples-dir\demo.db
UID=DBA
PWD=sql
```

有关 *samples-dir* 的信息，请参见“[示例目录](#)”一节第 368 页。

使用 ENG 连接参数

建议在使用嵌入式数据库时使用 ServerName (ENG) 连接参数。这样可确保当在同一计算机上有其它应用程序也在运行 SQL Anywhere 数据库服务器时，数据库能够连接到正确的数据库服务器。

使用 StartLine [START] 连接参数

以下连接参数显示如何自定义作为嵌入式数据库的示例数据库的启动。如果您希望使用一些选项（如高速缓存大小），这将会非常有用：

```
START=dbeng11 -c 8M
DBF=samples-dir\demo.db
UID=DBA
PWD=sql
```

影响服务器启动方式的连接参数有很多。建议您使用以下连接参数，而不要在 StartLine (START) 连接参数中提供相应的服务器选项：

- ServerName (ENG)
- DatabaseFile (DBF)
- DatabaseSwitches (DBS)
- DatabaseName (DBN)

使用 ELEVATE 连接参数

如果在 Windows Vista 上自动启动数据库服务器，则必须在连接字符串中指定 ELEVATE=YES，以便提升自动启动的数据库服务器的可执行能力。在 Windows Vista 上，只有提升的数据库服务器才能使用 AWE 内存或作为管理员用户来调用过程。

另请参见

- [“DatabaseFile 连接参数 \[DBF\]”](#) 一节第 258 页
- [“ServerName 连接参数 \[ENG\]”](#) 一节第 280 页
- [“StartLine 连接参数 \[START\]”](#) 一节第 281 页
- [“Elevate 连接参数”](#) 一节第 264 页
- [“打开 \[连接\] 窗口”](#) 一节第 87 页
- [“示例 SQL Anywhere 数据库连接”](#) 一节第 116 页

使用数据源连接

您可以在[数据源](#)中保存多组连接参数。所有 SQL Anywhere 接口（Open Client 和 jConnect 除外）都可以使用数据源。

◆ 使用数据源（Sybase Central 或 Interactive SQL）连接

1. 启动 Sybase Central 或 Interactive SQL。

如果未出现 [连接] 窗口：

- 在 Sybase Central 中，选择 [连接] » [使用 SQL Anywhere 11 连接]。
 - 在 Interactive SQL 中，选择 [SQL] » [连接]。
2. 单击 [标识] 选项卡。
 3. 在 [用户 ID] 字段中键入用户名。
 4. 在 [口令] 字段中，键入数据库的口令。
 5. 执行下列操作之一：
 - 单击 [ODBC 数据源名称]，然后输入引用 Windows 注册表中的数据源的 DataSourceName (DSN) 连接参数。单击 [浏览] 可查看数据源列表。
 - 单击 [ODBC 数据源文件]，然后输入引用保存在文件中的数据源的 FileDataSourceName (FILEDSN) 连接参数。单击 [浏览] 可查看文件列表。

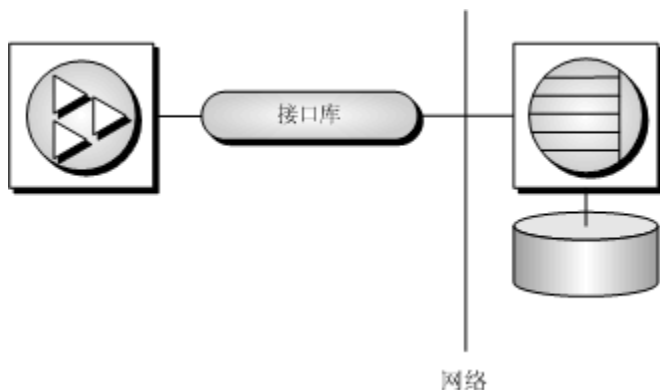
另请参见

- “打开 [连接] 窗口” 一节第 87 页
- “示例 SQL Anywhere 数据库连接” 一节第 116 页
- “在 Unix 上使用 ODBC 数据源” 一节第 95 页

连接到网络上的服务器

当连接到运行于本地网络或广域网上的网络服务器上的数据库时，客户端软件必须找到该数据库服务器并与之连接。SQL Anywhere 提供一个网络库，用于处理该任务。

网络连接建立在网络协议之上。所有平台都可以使用 TCP/IP。



指定服务器

对于给定的网络协议，SQL Anywhere 服务器名在本地域上必须唯一。以下示例将连接到在网络上运行的服务器：

```
ENG=svr-name  
DBN=db-name  
UID=user-id  
PWD=password  
CommLinks=all
```

指定 CommLinks=all 时，客户端库会先搜索具有给定名称的个人服务器，然后在网络上搜索具有给定名称的服务器。请参见“[CommLinks 连接参数 \[LINKS\]](#)”一节第 254 页。

指定协议

要改善性能，您可以指示网络库使用哪些协议。以下参数使用 TCP/IP 协议：

```
ENG=svr-name  
DBN=db-name  
UID=user-id  
PWD=password  
CommLinks=tcPIP
```

网络库通过在网络上进行广播来搜索服务器，该过程可能非常耗时。网络库找到服务器后，客户端库会将其名称和网络地址存储在一个文件 (*sasrv.ini*) 中，以后使用指定协议尝试与该服务器连接时，可再次使用该条目。随后的连接通常要比通过广播实现的连接快。

在缺省情况下，Sybase Central 和 Interactive SQL 中所有的网络连接都使用 TCP/IP 网络协议。

◆ 连接到网络服务器上的数据库（Sybase Central 或 Interactive SQL）

1. 启动 Sybase Central 或 Interactive SQL。

如果未出现 [连接] 窗口：

- 在 Sybase Central 中，选择 [连接] » [使用 SQL Anywhere 11 连接]。
- 在 Interactive SQL 中，选择 [SQL] » [连接]。

2. 单击 [标识] 选项卡。
3. 在 [用户 ID] 字段中键入用户名。
4. 在 [口令] 字段中，键入数据库的口令。
5. 单击 [数据库] 选项卡。
6. 在 [服务器名] 字段中，键入服务器名或单击 [查找]。
7. 在 [数据库名] 字段中，键入数据库的名称。
8. 单击 [确定]。

另请参见

- “客户端/服务器通信” 第 133 页
- “网络协议选项” 一节第 284 页
- “穿过防火墙进行连接” 一节第 136 页
- “打开 [连接] 窗口” 一节第 87 页
- “示例 SQL Anywhere 数据库连接” 一节第 116 页

使用缺省连接参数

您可以不对连接参数进行指定，而是使用缺省行为来建立连接。然而，如果该应用程序与其它 SQL Anywhere 应用程序一起安装，则在生产环境中使用缺省行为可能会导致问题。有关缺省行为的详细信息，请参见“对连接进行故障排除”一节第 124 页。

缺省数据库服务器和数据库

可使用缺省参数连接到一台装载了一个数据库的个人服务器：

```
UID=user-id  
PWD=password
```

缺省数据库服务器

如果一台个人服务器上装载了多个数据库，则可使用缺省服务器设置，并指定要连接的数据库：

```
DBN=db-name  
UID=user-id  
PWD=password
```

缺省数据库

如果正在运行多个服务器，则需指定要连接哪个服务器。如果该服务器上只装载了一个数据库，则无需指定数据库名。以下连接字符串使用缺省数据库连接到指定的服务器：

```
ENG=server-name  
UID=user-id  
PWD=password
```

本地服务器没有缺省值

以下连接字符串使用指定的数据库连接到指定的本地服务器：

```
ENG=server-name  
DBN=db-name  
UID=user-id  
PWD=password
```

网络服务器没有缺省值

连接到在另一台计算机上运行的网络服务器：

```
ENG=server-name  
DBN=dbn  
UID=user-id  
PWD=password  
CommLinks=tcPIP
```

如果未指定 CommLinks，则只尝试本地共享内存连接。

如果从 Sybase Central、Interactive SQL 或 SQL Anywhere 控制台实用程序 (dbconsole) 进行连接，您可以选择 [连接] 窗口上的 [搜索网络中的数据库服务器] 选项尝试网络连接。

通过 SQL Anywhere 实用程序连接

所有 SQL Anywhere 数据库实用程序都使用嵌入式 SQL 与服务器进行通信。

数据库实用程序如何获取连接参数值

许多管理实用程序都通过以下方法来获取连接参数值：

1. 使用在命令行上指定的值。例如，以下命令使用用户 ID DBA 和口令 sql 在缺省服务器上启动对缺省数据库的备份操作：

```
dbbackup -c "UID=DBA;PWD=sql" c:\backup
```

有关每个数据库实用程序的选项的详细信息，请参见“[数据库管理实用程序](#)”第 693 页。

2. 如果缺少任何值，则使用 SQLCONNECT 环境变量设置。SQL Anywhere 不自动设置该变量。请参见“[SQLCONNECT 环境变量](#)”一节第 362 页。

对连接进行故障排除

了解 SQL Anywhere 如何建立连接可以帮助您解决连接问题。有关网络特定问题（包括越过防火墙建立连接）的信息，请参见“客户端/服务器通信”第 133 页。

要建立连接，SQL Anywhere 执行以下步骤：

- 查找接口库
- 汇编一系列连接参数
- 查找服务器
- 查找数据库
- 在未找到数据库服务器时启动个人服务器

对于以下应用程序来说，SQL Anywhere 的连接过程是相同的：

- 使用 `SQLDriverConnect` 函数的任何 **ODBC 应用程序**，它是 ODBC 应用程序的常见连接方法。许多应用程序开发系统（如 Sybase PowerBuilder）属于这类应用程序。`SQLConnect` 函数也适用于 ODBC 应用程序。
- 使用 **嵌入式 SQL** 和使用推荐的函数 (`db_string_connect`) 来连接数据库的任何客户端应用程序。此外，SQL `CONNECT` 语句也可以用于嵌入式 SQL 应用程序和 Interactive SQL 中。它有两种形式：`CONNECT AS ...` 和 `CONNECT USING`。所有数据库管理工具（包括 Interactive SQL）都使用 `db_string_connect`。
- 任何使用 ADO DB Connection 对象的 **ADO 应用程序**。Provider 属性用于定位 OLE DB 驱动程序。Connection String 属性可以用 **DataSource** 替代 **DataSourceName**，用 **User ID** 替代 **UserID**。
- 使用 **iAnywhere JDBC 驱动程序** 将后面跟有标准连接字符串的 URL `jdbc:iAnywhere:` 作为参数传递给 `Driver Manager.getConnection` 方法的任何应用程序。该连接字符串必须包含 **DataSource=**，并指定一个 SQL Anywhere 数据源，或包含 **Driver=SQL Anywhere 11**（此参数在 Linux 和 Unix 上指定为 **Driver=libdbodbc11.so**）。

另请参见

- “对服务器启动进行故障排除”一节第 68 页
- “网络通信故障排除”一节第 143 页

查找接口库

通常，该 DLL 或共享库的位置对于用户是透明的。

ODBC 驱动程序的位置

对于 ODBC，接口库也称作 ODBC 驱动程序。ODBC 客户端应用程序调用 ODBC 驱动程序管理器，然后驱动程序管理器会查找 SQL Anywhere 驱动程序。

ODBC 驱动程序管理器在提供的数据源中搜索该驱动程序。当您使用 [ODBC 管理器] 或 `dbdsn` 实用程序创建数据源时，SQL Anywhere 将填写 ODBC 驱动程序的当前位置。数据源信息存储在 Windows 注册表或 Unix 系统信息文件中（缺省情况下名为 `.odbc.ini`）。

[嵌入式 SQL] 接口库的位置

嵌入式 SQL 应用程序按名称调用接口库。SQL Anywhere 嵌入式 SQL 接口库的名称如下：

- **Windows** `dblib11.dll`
- **Unix** `libdblib11`，扩展名特定于操作系统

OLE DB 驱动程序位置

提供程序名称 (SAOLEDB) 用于根据注册表中的条目查找 SQL Anywhere OLE DB 提供程序 DLL (`dboledb11.dll`)。这些项在安装或重新注册 SAOLEDB 提供程序时创建。

ADO.NET

ADO.NET 程序添加了对 SQL Anywhere ADO.NET 提供程序的引用，该提供程序名为 `iAnywhere.Data.SQLAnywhere.dll`。安装 .NET 数据提供程序 DLL 时，它被添加到 .NET 全局程序集高速缓存 (GAC) 中。

iAnywhere JDBC 驱动程序位置

运行应用程序时，Java 包 `jodbc.jar` 必须位于类路径中。系统必须能够找到本地 DLL 或共享对象。

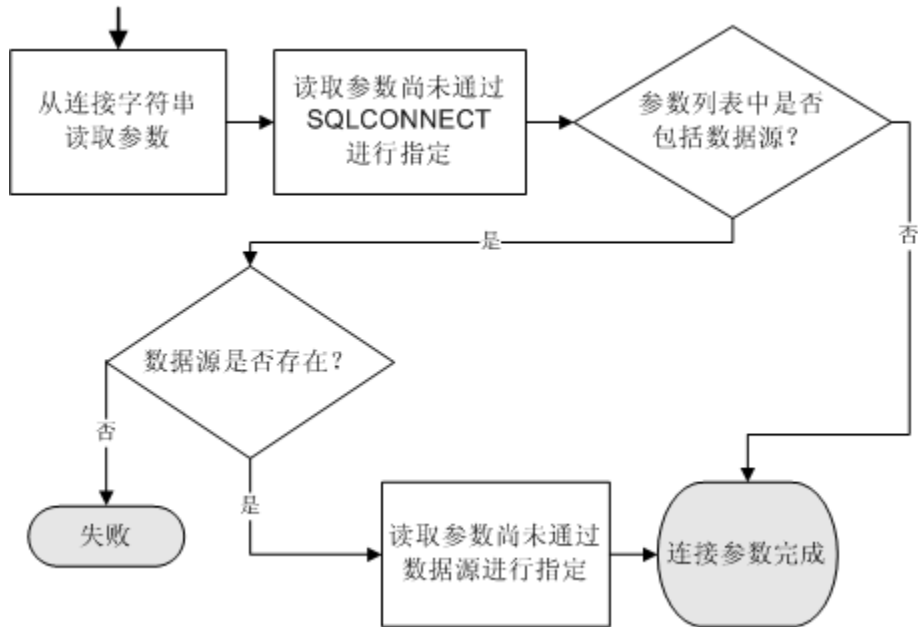
- **PC 操作系统** 在 PC 操作系统（如 Windows）上，会在当前目录、系统路径以及 `Windows` 和 `Windows\system32` 目录中搜索文件。
- **Unix 操作系统** 在 Unix 上，会搜索系统路径和用户库路径。

找到接口库之后

当客户端应用程序找到连接字符串时，会将其发送到接口库。接口库将使用该字符串汇集一系列连接参数并建立服务器连接。

汇集一系列连接参数

下图阐释了接口库如何汇集一系列连接参数以及如何建立连接。



● **优先** 保存在多个位置的参数具有下面的优先顺序：

1. 连接字符串
2. SQLCONNECT
3. 数据源

如果数据源和连接字符串中均提供了参数，则连接字符串的值将替换数据源的值。

● **失败** 只有当您在连接字符串或 SQLCONNECT 中指定了不存在的数据源时，才会在此阶段出现失败。

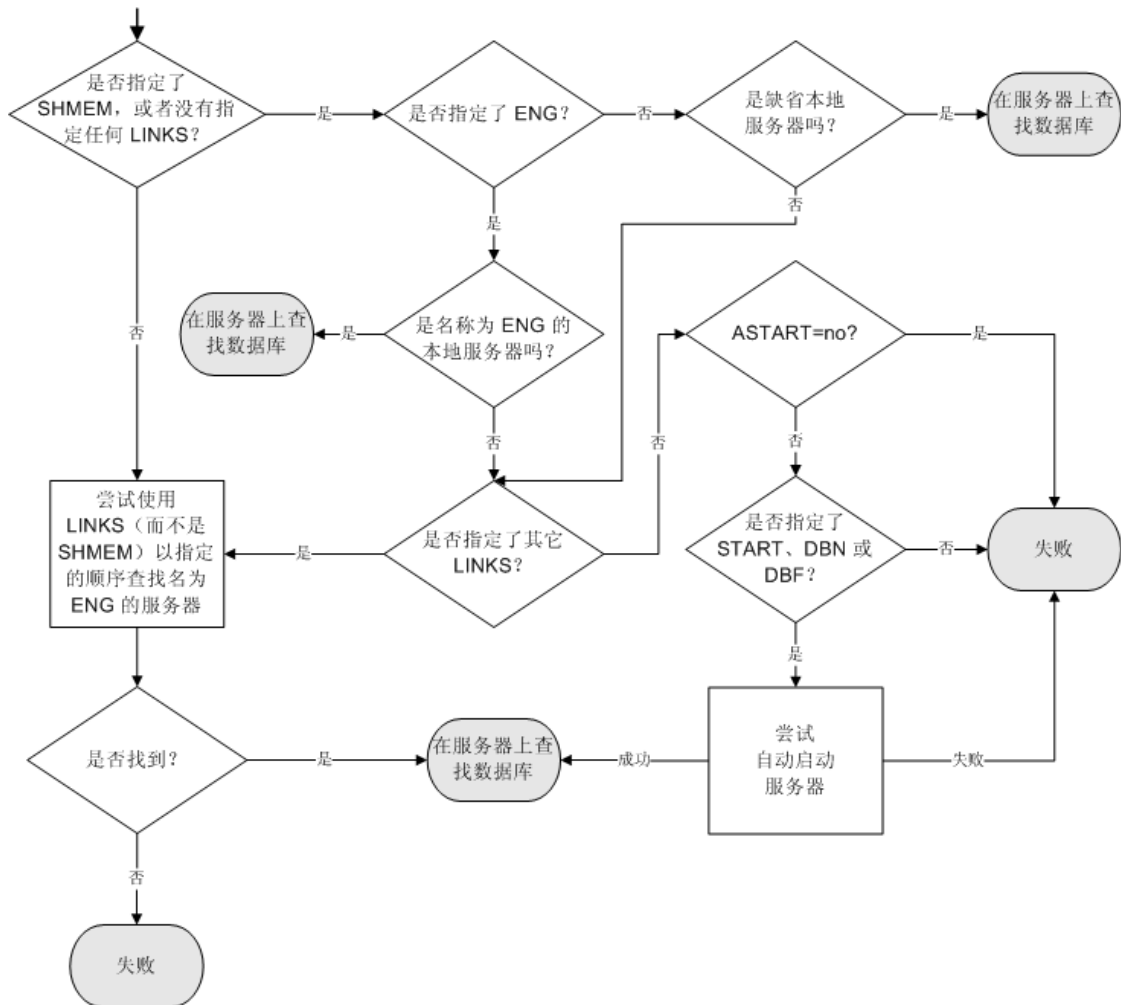
● **公用参数** 根据已在使用的其它连接，有些连接参数可以被忽略，包括：

- **AutoStop** 如果数据库已被装载，则被忽略。
- **DatabaseFile** 如果指定了 DatabaseName，并且具有此名称的数据库正在运行，则将忽略此参数。

接口库使用连接参数的完整列表来试图建立连接。

查找数据库服务器

SQL Anywhere 会搜索在 ServerName (ENG) 连接参数中指定的服务器名。如果未使用 ServerName (ENG) 连接参数并且未指定 CommLinks (LINKS) 连接参数，或者已经指定 CommLinks (LINKS) 连接参数并且包含共享内存，则 SQL Anywhere 将搜索缺省服务器。



如果 SQL Anywhere 找到一个服务器，它将尝试查找或装载该服务器所需的数据库。请参见“[查找数据库](#)”一节第 128 页。

如果 SQL Anywhere 不能找到服务器，则它会尝试启动个人服务器，这取决于连接参数。

注意

- 对于本地连接，查找服务器非常简单。对于网络连接，您可以使用 CommLinks (LINKS) 连接参数提供多个网络协议选项，从而采取多种方法调整搜索。
- 您可以在 CommLinks (LINKS) 连接参数的变量中为每个网络协议指定一组网络协议选项。
- 每次尝试查找服务器时，都包括两个步骤。首先，SQL Anywhere 在服务器名高速缓存中查找是否存在具有该名称的服务器（DoBroadcast 值为 none 时跳过此步骤）。然后，它使用可用的连接参数来尝试建立连接。
- 如果服务器是自动启动的，会使用 START、DBF、DBKEY、DBS、DBN、ENG 和 AUTOSTOP 连接参数的信息来构建自动启动的服务器的选项。

- 如果服务器有备用名称，则您只能使用备用服务器名称连接指定了备用服务器名称的数据库。您不能使用此备用服务器名称连接该数据库服务器上所运行的任何其它数据库。请参见“[-sn 数据库选项](#)”一节第 242 页。

使用广播转发器实用程序查找数据库服务器

广播转发器实用程序允许 SQL Anywhere 客户端穿过防火墙查找运行于其它子网中、UDP 广播通常无法到达的 SQL Anywhere 数据库服务器，而无需使用 HOST 连接参数或 LDAP。

◆ 使用广播转发器实用程序

1. 在子网中任一计算机上启动 DBNS（数据库名称服务）进程。
2. 在另一子网中的任意计算机上启动 DBNS 进程，并将第一台计算机的计算机名称或 IP 地址作为参数传递（使用 *address* 参数）。

这两个 DBNS 进程彼此建立一个 TCP/IP 连接。

3. DBNS 进程现在将监听其自身子网中的广播。每个 DBNS 进程通过 TCP/IP 连接将请求转发给另一方的 DBNS 进程，接收方进程在其子网中重新广播这些请求并将响应转发回源 DBNS 进程，源进程再将响应发送给其原始客户端。
4. 任一子网上的常规 SQL Anywhere 广播到达远程子网上的数据库服务器，客户端无需指定 HOST 参数即可连接到远程子网上的数据库服务器。

任意数量的 DBNS 进程都可以实现彼此之间的通信。每个 DBNS 进程会连接其所了解的另一 DBNS，不同的 DBNS 进程会共享其 DBNS 进程的列表。例如，假设您启动了 A 和 B 两个 DBNS 进程。如果在第三个子网中启动第三个 DBNS 进程 C，将 B 的地址传递给 C，则 B 会将 A 介绍给 C，然后 C 会连接到 A。

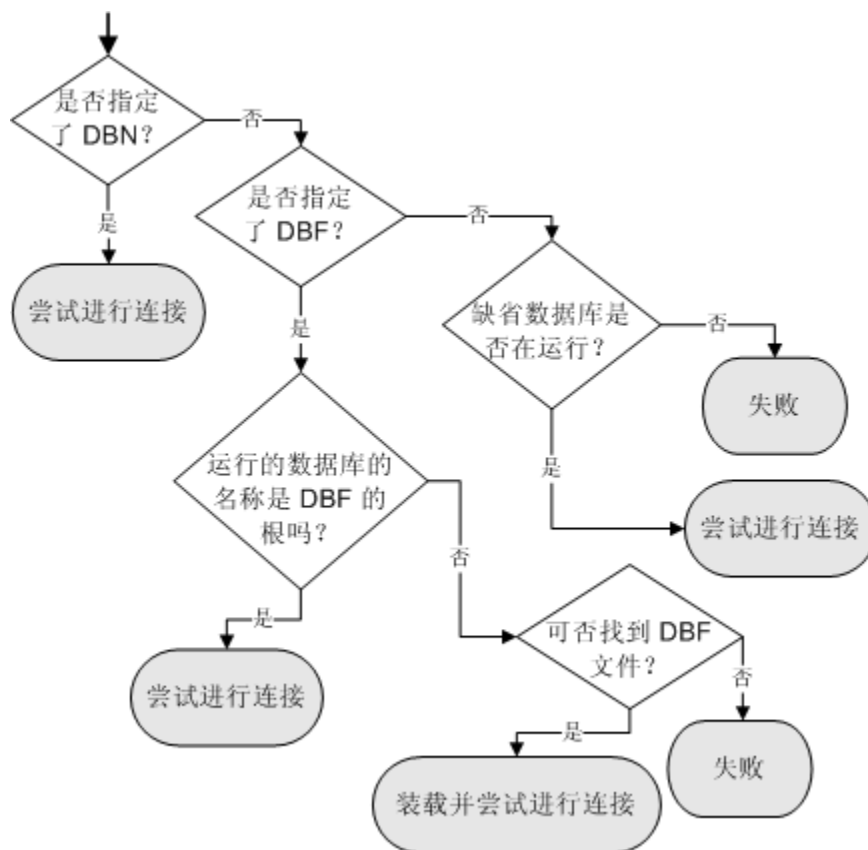
不必在单一子网中运行多个 DBNS 进程，也不建议这样做。

另请参见

- “[广播转发器实用程序 \(dbns11\)](#)”一节第 703 页

查找数据库

如果 SQL Anywhere 成功找到一个服务器，就会尝试查找数据库。例如：



高速缓存服务器名以提高连接速度

当 DoBroadcast (DOBROAD) 协议选项设置为 DIRECT 或 ALL 时，网络库通过使用 CommLinks (LINKS) 连接参数在网络上进行广播来查找网络上的数据库服务器。

调整广播

CommLinks (LINKS) 参数将列出待用协议的字符串视为参数，并且对于每个协议，也可将调整广播的各种网络协议选项视为参数。请参见“[网络协议选项](#)”一节第 284 页。

高速缓存服务器信息

通过在大型网络上广播来搜索具有特定名称的服务器可能非常耗时。高速缓存服务器地址将首次连接服务器时所基于的协议及其地址保存到一个文件中，并在以后连接时使用该信息，从而提高了网络连接速度。

服务器信息保存在名为 *sasrv.ini* 的高速缓存文件中。该文件包含几个部分，每一部分都采用如下形式：

```
[Server name]
LINKS=protocol_name
Address=address_string
```

在 Windows 上, *sasrv.ini* 的缺省位置是 *%ALLUSERSPROFILE%\Application Data\SQL Anywhere 11*, 而在 Unix 上为 *~/sqlanywhere11*。

注意

每个服务器都有唯一的名称是非常重要的。为不同服务器指定相同名称可导致标识问题。

如何使用高速缓存

如果高速缓存中的服务器名称和协议与连接字符串相匹配, 则 SQL Anywhere 会先尝试使用高速缓存地址进行连接。如果连接失败, 或者如果高速缓存中的服务器名和协议与连接字符串不匹配, 则使用连接字符串信息通过广播搜索服务器。如果广播成功, 则会覆盖高速缓存中的服务器名条目。如果找不到服务器, 则会删除高速缓存中的服务器名条目。如果将 *DoBroadcast* 协议选项设置为 *none*, 则会忽略所有高速缓存地址。

Interactive SQL 连接

发出 *CONNECT* 语句时, 如果已经连接了数据库, *Interactive SQL* 的行为会与嵌入式 SQL 的缺省行为有所不同。如果未在 *CONNECT* 语句中指定数据库或服务器, *Interactive SQL* 就会连接当前数据库, 而不是连接缺省数据库。该行为是数据库重装操作所必需的。请参见“[CONNECT 语句 \[ESQL\] \[Interactive SQL\]](#)”一节《[SQL Anywhere 服务器 - SQL 参考](#)》。

测试是否能找到服务器

可使用 *dbping* 实用程序排除连接故障, 并确定具有特定名称的服务器在网络上是否可用。

dbping 实用程序将连接字符串作为一个选项。在缺省情况下, 该实用程序不会启动服务器, 并且只使用查找服务器时所需的信息。将 *-d* 选项与 *dbping* 实用程序一起使用可启动服务器。

示例

以下命令行测试是否可通过 TCP/IP 连接使用名为 Waterloo 的服务器:

```
dbping -c "ENG=Waterloo;CommLinks=tcPIP"
```

以下命令用于测试是否可以在当前计算机上使用缺省服务器。

```
dbping
```

另请参见

- “[Ping 实用程序 \(dbping\)](#)”一节第 757 页

测试嵌入式 SQL 连接的性能

您可以使用 *Ping* 实用程序 (*dbping*) 并指定 *-s* 或 *-st* 选项来获取与嵌入式 SQL 连接的性能有关的信息。将收集以下统计信息:

统计信息	说明
DBLib 连接和断开	执行一次 DBLib 连接和断开所需的时间。注意，使用其它接口（如 ODBC）时，连接和断开的速度通常要比使用 DBLib 慢，因为完成连接所需的请求更多。
往返简单请求	从客户端发送请求到服务器所用的时间加上从服务器发回响应到客户端所需的时间。往返时间是平均等待时间的两倍。
发送吞吐量	该吞吐量根据每次迭代从 dbping 传输 100 KB 的数据到数据库服务器来计算。
接收吞吐量。	该吞吐量根据每次迭代从数据库服务器传输 100 KB 的数据到 dbping 来计算。

如果网络的往返时间和吞吐量都高，则由于往返耗费的时间较高，报告的吞吐量将比实际网络吞吐量要低。使用 `dbping -s` 对于指示通信压缩是否会改善性能很有帮助。性能统计信息是近似的结果，当客户端和服务器端计算机均完全闲置时，得到的统计结果会更精确。如果使用通信压缩，则传输的数据大约可以压缩到其原始大小的 25%。

下面是针对 `dbping` 命令 `dbping -s -c`

"UID=DBA;PWD=sql;ENG=sampleserver;LINKS=TCPIP" 的 `dbping -s` 输出示例：

```
SQL Anywhere 服务器强制回应实用程序版本 11.0.1.1658
已在地址 10.25.107.108 处连接到 SQL Anywhere 11.0.1.1657 服务器 "sampleserver"
和数据库 "sample"。
性能统计信息
```

性能统计信息	次数	总时间	平均时间
DBLib 连接或断开连接	175 次	1024 毫秒	5 毫秒
往返简单请求	2050 次请求	1024 毫秒	<1 毫秒
发送量	7600 KB	1024 毫秒	7421 KB/秒
接收量	10100 KB	1024 毫秒	9863 KB/秒

```
强制数据库回应成功。
```

另请参见

- “Ping 实用程序 (dbping)” 一节第 757 页

从数据库断开连接

要断开用户与数据库的连接，请参见“管理连接的用户”一节第 442 页。

◆ 从数据库断开连接 (Sybase Central)

1. 选择一个数据库。
2. 选择 [文件] » [断开连接]。

◆ 从数据库断开连接 (SQL)

- 执行 DISCONNECT 语句。

请参见“DISCONNECT 语句 [ESQL] [Interactive SQL]”一节《SQL Anywhere 服务器 - SQL 参考》和“DROP CONNECTION 语句”一节《SQL Anywhere 服务器 - SQL 参考》。

◆ 断开其他用户与数据库的连接 (SQL)

1. 以具有 DBA 权限的用户身份连接到数据库。
2. 使用 sa_conn_info 系统过程来确定要断开连接的用户的连接 ID。
3. 执行 DROP CONNECTION 语句。

示例

下面的语句说明如何在 Interactive SQL 中使用 DISCONNECT 语句断开当前连接 conn1:

```
DISCONNECT conn1;
```

下面的语句说明如何在嵌入式 SQL 中使用 DISCONNECT:

```
EXEC SQL DISCONNECT :conn-name
```

下面的语句会删除 ID 号为 4 的连接。

```
DROP CONNECTION 4;
```

客户端/服务器通信

目录

支持的网络协议	134
使用 TCP/IP 协议	135
调整通信压缩设置以改善性能	141
网络通信故障排除	143

支持的网络协议

正确配置的 SQL Anywhere 数据库服务器可以在以下网络和协议上运行：

- **Windows (Windows 2003 Server 除外)** TCP/IP 协议。
- **Windows 2003 Server (32 位)** TCP/IP 协议。
- **Windows 2003 Server (64 位)** TCP/IP 协议。
- **Windows Mobile** TCP/IP 协议。
- **Unix** TCP/IP 协议。

每个平台的客户端库与对应服务器所支持的协议是相同的。为使 SQL Anywhere 能够正常运行，在客户端和服务计算机上都必须正确地安装和配置网络协议 (TCP/IP)。

使用 TCP/IP 协议

TCP/IP 是一组协议，随着 Internet 和万维网的扩展，这组协议得到了广泛的应用。

UDP 是一种 IP 之上的传送层协议。SQL Anywhere 使用 IP 之上的 UDP 进行初始服务器名称解析，之后使用 TCP 进行连接和通信。

使用 TCP/IP 协议时，可以使用传送层安全和 ECC 或 RSA 加密技术来保护客户端/服务器通信的安全。

请参见“[传送层安全](#)”第 1031 页。

SQL Anywhere 中的 IPv6 支持

在启用了 IPv6 的计算机上，缺省情况下，网络数据库服务器会监听所有的 IPv6 和 IPv4 地址。Windows、Linux、Mac OS X、Solaris、AIX 和 HP-UX 上都支持 IPv6。

在大多数情况下，不需要更改服务器启动行便可使用 IPv6。在需要指定 IP 地址的情况下，服务器和客户端库都接受 IPv4 和 IPv6 地址。例如，如果计算机上启用了多个网卡，则其可能有两个 IPv4 地址和两个 IPv6 地址。如果想要让数据库服务器只监听其中一个 IPv6 地址，则可以指定以下格式的地址：

```
dbsrv11 -x tcpip(MyIP=fd77:55f:5a64:52a:202:5445:5245:444f) ...
```

类似地，如果客户端应用程序需要指定服务器的 IP 地址，则连接字符串或 DSN 可以包含该地址，格式如下所示：

```
...;LINKS=tcpip(HOST=fe80::5445:5245:444f);...
```

每个接口都指定有一个接口标识符，该标识符出现在 IPv6 地址的结尾。例如，如果 *ipconfig.exe* 列出地址 `[fe80::5445:5245:444f%7]`，则接口标识符为 7。在 Windows 平台上指定 IPv6 地址时，应使用该接口标识符。在 Unix 上，可指定接口标识符或接口名称（接口名称是 *ifconfig* 所报告的接口的名称）。例如，在以下 IPv6 地址中，接口名称为 `eth1`：`[fe80::5445:5245:444f%eth1]`。在 Linux（内核 2.6.13 及更高版本）上指定 IPv6 地址时，接口标识符是必需的。此要求会影响以下协议选项指定的值：

- 广播
- 主机
- MyIP

例如，假设 *ipconfig.exe* 列出了两个接口，一个标识符为 1，另一个标识符为 2。如果要查找网络上 2 号接口使用的数据库服务器，则可以指示客户端库仅在该接口上进行广播：

```
LINKS=tcpip(BROADCAST=ff02::1%2)
```

请注意，`[ff02::1]` 是 IPv6 链路本地多路广播地址。

另请参见

- “Broadcast 协议选项 [BCAST]” 一节第 285 页
- “Host 协议选项 [IP]” 一节第 292 页
- “MyIP 协议选项 [ME]” 一节第 301 页

在 Windows 上使用 TCP/IP

所有 Windows 平台上数据库服务器的 TCP/IP 实现都使用 Winsock 2.2。Windows Mobile 上的客户端使用 Winsock 1.1 标准。

如果尚未安装 TCP/IP，可以通过双击 [控制面板] 中的 [网络设置] 安装 TCP/IP 协议。

调整 TCP/IP 性能

增加数据包大小可以改善查询响应时间，对在客户端与服务器进程之间传输大量数据的查询而言尤其如此。可以通过两种方法设置数据包大小，一种是在数据库服务器命令中使用 `-p` 选项，一种是在连接配置文件中设置 `CommBufferSize (CBSIZE)` 连接参数。

另请参见

- “`-p` 服务器选项” 一节第 200 页
- “`CommBufferSize` 连接参数 [CBSIZE]” 一节第 253 页

穿过防火墙进行连接

当客户端应用程序在防火墙的一边，而服务器在另一边时，连接会受到限制。防火墙软件按照网络端口过滤网络数据包。而且，通常不允许 UDP 数据包穿过防火墙。

在穿过防火墙进行连接时，您必须在应用程序连接字符串的 `CommLinks (LINKS)` 连接参数中使用一组协议选项。

- **Host** 将此参数设置为运行数据库服务器的主机名。可以使用其简写形式 `IP`。
- **ServerPort** 如果数据库服务器未使用缺省端口 2638，则必须指定它所使用的端口。可以使用其简写形式 `Port`。
- **ClientPort** 将此参数设置为供客户端应用程序使用的某一范围的允许值。您可以使用此参数的简写形式 `CPort`。此选项取决于防火墙的配置，可能并不需要。
- **DoBroadcast=NONE** 设置此参数可防止在连接服务器时使用 UDP。

必须对防火墙进行配置，使其允许在 SQL Anywhere 服务器地址与所有 SQL Anywhere 客户端地址之间进行 TCP/IP 通信。SQL Anywhere 服务器的地址由运行 SQL Anywhere 服务器的计算机的 IP 地址 (`HOST` 参数) 和 SQL Anywhere 服务器的 IP 端口号 (`ServerPort` 协议选项，缺省值为 2638) 组成。每个 SQL Anywhere 客户端的地址都是由客户端计算机的 IP 地址和客户端 IP 端口范围 (`ClientPort` 协议选项) 组成。在最简单的配置中，允许开放所有客户端端口。如果只允许开放特定

客户端端口，则请指定一个端口数超过每个客户端计算机最大并发连接数的端口范围，因为几分钟超时过后才可以再次使用客户端端口。

请参见“[ClientPort 协议选项 \[CPORT\]](#)”一节第 290 页。

示例

以下连接字符串片段将客户端应用程序限制为只能使用端口 5050 到 5060，并通过服务器端口 2020 连接到地址为 myhost 的计算机上正在运行的名为 myeng 的服务器。由于 DoBroadcast 选项的设置，不执行 UDP 广播。

```
ENG=myeng;LINKS=tcip(ClientPort=5050-5060;HOST=myhost;PORT=2020;DoBroadcast=NONE)
```

另请参见

- “[CommLinks 连接参数 \[LINKS\]](#)”一节第 254 页
- “[ClientPort 协议选项 \[CPORT\]](#)”一节第 290 页
- “[ServerPort 协议选项 \[PORT\]](#)”一节第 303 页
- “[Host 协议选项 \[IP\]](#)”一节第 292 页
- “[DoBroadcast 协议选项 \[DOBROAD\]](#)”一节第 291 页

在拨号网络连接上进行连接

您可以使用连接和协议选项，以协助通过拨号链接连接到数据库。

在客户端，您应该指定以下协议选项：

- **Host 参数** 您应该使用 Host (IP) 协议选项指定数据库服务器的主机名或 IP 地址。请参见“[Host 协议选项 \[IP\]](#)”一节第 292 页。
- **DoBroadcast 参数** 如果指定 Host (IP) 协议选项，则无需执行广播搜索来查找数据库服务器。由于此原因，请使用直接广播。请参见“[DoBroadcast 协议选项 \[DOBROAD\]](#)”一节第 291 页。
- **MyIP 参数** 应在客户端上设置 [**MyIP=NONE**]。请参见“[MyIP 协议选项 \[ME\]](#)”一节第 301 页。
- **TIMEOUT 参数** 设置 TIMEOUT (TO) 协议选项以增加客户端在搜索服务器时将等待的时间。请参见“[Timeout 协议选项 \[TO\]](#)”一节第 306 页。

典型的 CommLinks (LINKS) 连接参数可能会是这样的：

```
LINKS=tcip(MyIP=NONE;DoBroadcast=DIRECT;HOST=server_ip)
```

对通过 TCP/IP 的客户端/服务器通信进行加密

缺省情况下，通信包是不加密的，但这样可能会引起潜在的安全风险。您可以使用简单加密技术或传送层安全来保护客户端应用程序和数据库服务器之间在 TCP/IP 上的通信安全。传送层安全提供了服务器验证、采用 ECC 或 RSA 加密技术的高度加密以及其它用于保护数据完整性的功能。

请参见“[传送层安全](#)”第 1031 页。

使用 LDAP 服务器进行连接

如果使用的是 Windows（Windows Mobile 除外）或 Unix 平台，则可以指定一个中央 LDAP 服务器来跟踪企业中的所有数据库服务器。如果数据库服务器在 LDAP 服务器中自行注册，客户端便可以查询 LDAP 服务器，找到其要查找的数据库服务器，无论这些服务器是在 WAN、LAN 上还是位于防火墙的后面。客户端不需要指定 IP 地址 (HOST=)。服务器枚举实用程序 (dblocate) 也可以使用 LDAP 服务器来查找其它同类服务器。

LDAP 仅与 TCP/IP 一起使用，且仅在网络数据库服务器上使用。

在 AIX 上配合使用 SQL Anywhere 与 LDAP 服务器

要配合使用 SQL Anywhere 11 与 AIX 6，必须在 `/usr/lib` 中创建链接，或者确保具有 LDAP 库的目录包括在 LIBPATH 中，以确保能够找到 LDAP 系统库。

◆ 在 `/usr/lib` 中创建链接

- 以 root 用户身份运行以下命令：

```
cd /usr/lib
ln -s /opt/IBM/ldap/V6.1/lib64/libibmldap.a libibmldap64.a
ln -s /opt/IBM/ldap/V6.1/lib/libibmldap.a
```

◆ 将包含 LDAP 库的目录添加到 LIBPATH

1. 以 root 用户身份运行以下命令，在 `/usr/lib` 下创建链接：

```
cd /usr/lib
ln -s /opt/IBM/ldap/V6.1/lib64/libibmldap.a libibmldap64.a
ln -s /opt/IBM/ldap/V6.1/lib/libibmldap.a
```

2. 确保具有 LDAP 库的目录位于 LIBPATH 中。

例如，对于 64 位库：

```
export LIBPATH=/opt/IBM/ldap/V6.1/lib64:$LIBPATH
```

例如，对于 32 位库：

```
export LIBPATH=/opt/IBM/ldap/V6.1/lib:$LIBPATH
```

配置 `saldap.ini` 文件

要启用此功能，必须在数据库服务器计算机和每台客户端计算机上都创建一个文件，其中包含有关如何查找和连接到 LDAP 服务器的信息。此文件的缺省名称是 `saldap.ini`，但可以对其进行配置。如果此文件不存在，将以静默方式禁用 LDAP 支持。

如果未使用 LDAP 参数指定完整路径，则该文件必须位于放置 SQL Anywhere 可执行文件的目录（例如，如果使用的是 Windows，则是 `install-dir\bin32`）中。该文件必须采用以下格式：

```
[LDAP]
server=computer-running-LDAP-server
port=port-number-of-LDAP-server
basedn=Base-DN
authdn=Authentication-DN
```

```
password=password-for-authdn
search_timeout=age-of-timestamps-to-be-ignored
update_timeout=frequency-of-timestamp-updates
read_authdn=read-only-authentication-domain-name
read_password=password-for-authdn
```

可以使用文件隐藏实用程序 (dbfhide) 来进行简单加密，以对 *saldap.ini* 文件的内容进行模糊处理。请参见“[文件隐藏实用程序 \(dbfhide\)](#)”一节第 724 页。

如果该文件的名称不是 *ldap.ini*，则必须使用 LDAP 参数指定文件名。

server 运行 LDAP 服务器的计算机的名称或 IP 地址。在 Unix 上，此值是必需的。在 Windows 上，如果缺少此条目，Windows 会查找在本地域控制器上运行的 LDAP 服务器。

port LDAP 服务器使用的端口号。缺省值为 389。

basedn 存储 SQL Anywhere 条目的子树的域名。此值缺省为树的根。

authdn 验证域名。该域名必须是 LDAP 目录中对 **basedn** 拥有写权限的一个现有用户对象。对于数据库服务器，此参数是必需的，但在客户端上将被忽略。

password **authdn** 的口令。对于数据库服务器，此参数是必需的，但在客户端上将被忽略。

search_timeout 时间戳的有效期限，到期时客户端和/或服务器枚举实用程序 (dblocate) 将忽略时间戳。值 0 将禁用此选项，此时将假定所有条目都处于最新状态。缺省值为 600 秒（10 分钟）。

update_timeout LDAP 目录中更新时间戳的周期。值为 0 时将禁用此选项，这样数据库服务器就永远不会更新时间戳。缺省值为 120 秒（2 分钟）。

read_authdn 只读验证域名。该域名必须是 LDAP 目录中对 **basedn** 拥有读权限的一个现有用户对象。只有在 LDAP 服务器要求进行非匿名绑定才能执行搜索时，此参数才是必需的。例如，如果将 Active Directory 用作 LDAP 服务器，则此字段通常是必填字段。如果缺少此参数，绑定将是匿名的。

read_password **authdn** 的口令。仅当指定了 **read_authdn** 参数时，此参数在客户端上才是必需的。

示例

以下是一个示例 *saldap.ini* 文件：

```
[LDAP]
server=ldapserver
basedn=dc=iAnywhere,dc=com
authdn=cn=SAserver,ou=iAnywhereASA,dc=iAnywhere,dc=com
password=secret
```

条目存储在名为 *iAnywhereASA* 的 **basedn** 的一个子树中。必须先创建此条目，SQL Anywhere 才能使用 LDAP。若要创建子树，可以使用 LDAPADD 实用程序，提供以下信息：

```
dn: ou=iAnywhereASA,basedn
objectClass: organizationalUnit
objectClass: top
ou: iAnywhereASA
```

服务器启动时会检查 LDAP 文件中是否已经存在同名条目。如果找到同名条目且满足以下条件之一，则会替换该同名条目：LDAP 中的位置条目与尝试启动的数据库服务器相符；LDAP 条目中的时间戳字段已超过 10 分钟的有效期限（该超时值是可配置的）。

如果这两个条目都不满足，则表明与尝试启动的数据库服务器同名的另一个数据库服务器正在运行，启动将会失败。

为确保 LDAP 中的条目是最新的，数据库服务器会每 2 分钟更新一次 LDAP 条目中的时间戳字段。如果一个条目的时间戳超过 10 分钟，客户端将忽略该 LDAP 条目。这两项设置都是可配置的。

在客户端上，将会先搜索 LDAP 目录，再进行任何广播，这样一来，如果找到了数据库服务器，就不会发送广播。LDAP 搜索的速度很快，因此失败时不会造成明显的延迟。

服务器枚举实用程序 (dblocate) 也使用 LDAP—LDAP 中列出的所有数据库服务器都将添加到返回的数据库服务器列表中。这使服务器枚举实用程序 (dblocate) 可以列出通常不会返回的数据库服务器，例如，那些广播无法到达的服务器。时间戳超过 10 分钟有效期限的条目不包括在内。

调整通信压缩设置以改善性能

在某些情况下，设置数据包的最小压缩大小并为一个或所有连接启用压缩可以提高 SQL Anywhere 的性能。

要确定在特定情况下启用压缩是否有帮助，建议先使用特定应用程序对特定网络进行性能分析，然后再于生产环境中使用通信压缩。性能结果将视所使用的网络类型、应用程序及所传输的数据而异。

最基本的压缩调整方法很简单，即在连接或服务器级别上启用或禁用 **Compression (COMP)** 连接参数。通过调整 **CompressionThreshold (COMPTH)** 连接参数，可以对压缩性能进行更高级的微调。

启用压缩会增加数据包中存储的信息量，从而减少了传输一组特定数据所需发送的数据包数量。通过减少数据包的数量，可以更快地传输数据。

有关性能分析的详细信息，请参见“性能监控器统计”一节《SQL Anywhere 服务器 - SQL 的用法》和“sa_conn_compression_info 系统过程”一节《SQL Anywhere 服务器 - SQL 参考》。

启用压缩

在某些情况下，为一个连接（或所有连接）启用压缩可以显著提高 SQL Anywhere 的性能，这些情况包括：

- 在某些无线网络、某些调制解调器、串行链接及某些 WAN 等慢速网络上使用时。
- 在具有内置压缩功能的慢速网络上与 SQL Anywhere 加密联合使用时，因为数据包在加密之前会先经过压缩。

不过，启用压缩有时也会导致性能下降。例如：

- 通信压缩会占用更多的内存和 CPU 资源。它可能会导致性能下降，对 LAN 和其它快速网络而言尤其如此。
- 大多数调制解调器和某些慢速网络都已具有内置压缩功能。在上述情况下，除非同时对数据进行加密，否则 SQL Anywhere 通信压缩可能不会给性能带来更大益处。

有关压缩的详细信息，请参见“Compress 连接参数 [COMP]”一节第 255 页和“-pc 服务器选项”一节第 200 页。

修改压缩阈值

也可以调整压缩阈值来提高 SQL Anywhere 的性能。对于大多数网络，不需要更改压缩阈值。

如果启用了压缩，则可能会压缩也可能不会压缩单个数据包，具体情况视其大小而定。例如，即使启用了通信压缩，SQL Anywhere 也不压缩小于压缩阈值的数据包。此外，小数据包（约小于 100 字节）通常根本不压缩。由于压缩数据包时需要 CPU 时间，因此尝试压缩小数据包实际上可能会降低性能。

一般来说，降低压缩阈值可能会使非常慢的网络的性能有所改善，而提高压缩阈值可能会通过减少 CPU 使用率来改善性能。但是，由于降低压缩阈值在客户端和服务器上均会增加 CPU 使用率，因此应进行性能分析以确定更改压缩阈值是否有好处。

请参见“[CompressionThreshold 连接参数 \[COMPTH\]](#)”一节第 256 页和“[-pt 服务器选项](#)”一节第 201 页。

◆ 调整 SQL Anywhere 压缩设置

1. 启用通信压缩。

使用高度可压缩的数据并使用较大的数据包大小的大数据传送，往往会获得最佳的压缩率。

有关启用压缩的详细信息，请参见“[Compress 连接参数 \[COMP\]](#)”一节第 255 页和“[-pc 服务器选项](#)”一节第 200 页。

2. 调整 CompressionThreshold 设置。

在速度很慢的网络上降低压缩阈值可能会改善性能，而提高压缩阈值可能会通过减少 CPU 占用来改善性能。

有关调整 CompressionThreshold (COMPTH) 连接参数的详细信息，请参见“[CompressionThreshold 连接参数 \[COMPTH\]](#)”一节第 256 页和“[-pt 服务器选项](#)”一节第 201 页。

网络通信故障排除

网络软件包括几个不同的组件，因而增大了出现问题的可能性。尽管这里提供了一些有关网络故障排除的技巧，但在进行网络故障排除时，主要还是要通过网络通信软件供应商提供的网络通信软件文档和技术支持来获得协助。

使用记录

如果指定 `-z` 数据库服务器选项，则会在数据库服务器消息窗口中显示作故障排除用途的通信诊断消息和其它消息。这些消息可以帮助您诊断连接失败的原因、连接尝试所使用的连接参数，以及所使用的通信链接。

确保正在使用兼容协议

确保客户端和数据库服务器使用相同的协议。服务器的 `-x` 选项会选择供服务器使用的一组协议，CommLinks (LINKS) 连接参数会为客户端应用程序执行相同的操作。

您可以使用这些选项来确保每个应用程序都在使用相同的协议。

缺省情况下，网络数据库服务器使用所有可用的协议。服务器在任何活动协议上都支持客户端请求。缺省情况下，客户端只使用共享内存协议。如果将 CommLinks (LINKS) 连接参数设置为 `all`，客户端就可以使用所有可用的协议。

另请参见

- “`-x` 服务器选项” 一节第 221 页
- “CommLinks 连接参数 [LINKS]” 一节第 254 页

确保具有最新的驱动程序

网络适配器驱动程序的版本较旧可能是发生通信问题的原因之一。您应该确保安装了最新版本的网络适配器驱动程序。通常可以从网卡制造商或供应商处获得最新的网络适配器驱动程序。

测试 TCP/IP 协议

`ping` 实用程序在测试 TCP/IP 的安装和配置是否正确上很有用。

使用 Ping 测试 IP 层

每个 IP 层都有一个关联的地址—对于 IPv4，该地址是以句点分隔的四个整数数字（如 191.72.109.12）。Ping 将 IP 地址作为参数，并尝试将单个数据包发送到该地址。

首先，通过使用 `ping` 实用程序检测自己的计算机配置得是否正确。如果 IP 地址为 191.72.109.12，可运行以下命令，并等待以确定是否真的发送了数据包：

```
ping 191.72.109.12
```

如果已经发送了数据包，输出内容将与以下所示内容类似：

```
Pinging 191.72.109.12 with 32 bytes of data:  
Reply from 191.72.109.12:bytes=32 time<.10ms TTL=32  
Reply from 191.72.109.12:bytes=32 time<.10ms TTL=32  
Reply from 191.72.109.12:bytes=32 time<.10ms TTL=32 ...
```

这意味着该计算机能够将数据包发送给它自己。这样就有理由确信 IP 层的设置正确。也可以要求其他运行 TCP/IP 的人提供其 IP 地址，然后尝试使用 ping 实用程序检测其计算机。

在进行下一步之前您应该确保可以从客户机对运行数据库服务器的计算机执行 Ping 命令并获得回应。

如果要尝试连接到 IPv6 网络上的某个主机，则必须先确保客户端计算机上安装了 IPv6。在 Windows XP 上，运行命令 [ipv6 install] 来安装 IPv6。缺省情况下，IPv6 安装在 Windows Vista 上。在每种 Unix 操作系统上的 IPv6 安装都不同；请查阅操作系统文档中有关启用 IPv6 的说明。

安装并启用 IPv6 后，即可使用 [ping6] 命令来执行以上所述的 ping 命令所执行的操作。例如：

```
ping6 fe80::213:ceff:fe24:ca6  
  
Pinging fe80::213:ceff:fe24:ca6  
from fe80::213:ceff:fe24:ca6%6 with 32 bytes of data:  
  
Reply from fe80::213:ceff:fe24:ca6%6:bytes=32 time<1ms  
Reply from fe80::213:ceff:fe24:ca6%6:bytes=32 time<1ms  
Reply from fe80::213:ceff:fe24:ca6%6:bytes=32 time<1ms ...
```

诊断布线问题

错误的网络布线或连接器故障可能会导致一些难以查明原因的问题。请尝试在相同配置下在相似的计算机上再现问题。如果问题只在一台计算机上出现，则可能是布线问题或硬件问题。

常见问题的检查表

下表列出一些常见问题及其解决办法。

有关对数据库或数据库服务器连接进行故障排除的信息，请参见“[对连接进行故障排除](#)”一节第 124 页和“[对服务器启动进行故障排除](#)”一节第 68 页。

如果在尝试进行连接时接收到 [未找到数据库服务器] 消息，则表明客户端无法在网络上找到该数据库服务器。检查是否存在以下问题：

- 在 TCP/IP 协议下，客户端通过广播请求的方式来搜索数据库服务器。这样的广播通常无法通过网关，因此将无法找到另一个（子）网络中计算机上的任何数据库服务器。如果是这种情况，则必须使用 HOST (IP) 协议选项来提供运行服务器的计算机的主机名。
- 客户端和服务器之间的防火墙可能会阻止连接。请参见“[穿过防火墙进行连接](#)”一节第 136 页。

- 个人服务器只接受来自同一台计算机的连接。如果客户端和服务端位于不同的计算机上，则必须使用网络服务器。
- 没有正确地安装网络驱动程序，或者没有正确地对网络布线。
- 如果接收到 [无法初始化请求的通信链接] 消息，则表明一个或多个链接无法启动。导致此问题的原因可能是尚未安装网络驱动程序。请参见网络文档，以了解如何安装您想要使用的驱动程序。
- 如果是通过 jConnect 进行连接，则服务器应使用 TCP/IP 协议。
- 如果要尝试连接到本地计算机上的数据库服务器，请确保将 [连接] 窗口中 [数据库] 选项卡上 [搜索网络中的数据库服务器] 选项的复选标记清除。如果要尝试连接到不在本地计算机上运行的数据库服务器，则可以选中此选项。

有关网络协议选项的详细信息，请参见“网络协议选项”一节第 284 页。

调整超时值

如果您正遇到连接意外断开的问题，请考虑调整活动超时值或空闲超时值。

另请参见

- “LivenessTimeout 连接参数 [LTO]”一节第 272 页
- “-tl 服务器选项”一节第 212 页
- “Idle 连接参数”一节第 268 页
- “-ti 服务器选项”一节第 212 页

数据库服务器

目录

SQL Anywhere 数据库服务器	148
数据库服务器选项	156
数据库选项	234

SQL Anywhere 数据库服务器

启动个人数据库服务器或网络数据库服务器。

语法

```
{ dbeng11 | dbsrv11 }
[ server-options ] [ database-file [ database-options ] ...]
```

服务器选项

服务器选项	说明
@data	从配置文件或环境变量中读入选项。请参见“@data 服务器选项”一节第 156 页。
-?	显示用法信息。请参见“-? 服务器选项”一节第 157 页。
-b	以批量操作模式运行。请参见“-b 服务器选项”一节第 157 页。
-c size	设置初始高速缓存大小。请参见“-c 服务器选项”一节第 158 页。
-ca 0	禁用动态调整高速缓存大小 [Windows、Unix、Mac OS X]。请参见“-ca 服务器选项”一节第 159 页。
-cc { + - }	收集要用于高速缓存预热的数据库页数的信息。请参见“-cc 服务器选项”一节第 160 页。
-ch size	设置高速缓存大小的上限 [Windows、Unix、Mac OS X]。请参见“-ch 服务器选项”一节第 161 页。
-cl size	设置高速缓存大小的下限 [Windows、Unix、Mac OS X]。请参见“-cl 服务器选项”一节第 162 页。
-cm size	指定为 Address Windowing Extensions (AWE) 高速缓存分配的地址空间量 [Windows]。请参见“-cm 服务器选项”一节第 163 页。
-cp location[;location ...]	指定在其中搜索类的一组目录或 jar 文件。请参见“-cp 服务器选项”一节第 164 页。
-cr { + - }	将数据库页预热高速缓存。请参见“-cr 服务器选项”一节第 165 页。
-cs	在数据库服务器消息窗口中显示高速缓存的使用情况。请参见“-cs 服务器选项”一节第 165 页。
-cv { + - }	控制是否在数据库服务器消息窗口中显示有关高速缓存预热的消息。请参见“-cv 服务器选项”一节第 166 页。

服务器选项	说明
-cw	允许使用 Address Windowing Extensions 来设置数据库服务器高速缓存的大小 [Windows]。请参见 “-cw 服务器选项” 一节第 167 页。
-dt temp-file-dir	指定临时文件的存储目录。请参见 “-dt 服务器选项” 一节第 170 页。
-ec encryption-options	启用包加密 [网络服务器]。请参见 “-ec 服务器选项” 一节第 171 页。
-ep	提示输入加密密钥。请参见 “-ep 服务器选项” 一节第 173 页。
-es	允许在共享内存上进行未加密的连接。请参见 “-es 服务器选项” 一节第 174 页。
-f	强制在没有事务日志的情况下启动数据库。请参见 “-f 恢复选项” 一节第 175 页。
-fc filename	指定包含文件系统已满回调函数的 DLL 的文件名。请参见 “-fc 服务器选项” 一节第 175 页。
-fips	要求使用 FIPS 认可的算法对数据库和通信进行高度加密 [Windows]。请参见 “-fips 服务器选项” 一节第 176 页。
-ga	在最后的非 HTTP 客户端连接关闭后自动卸载数据库。此外，还在最后的数据库关闭后关机。请参见 “-ga 服务器选项” 一节第 177 页。
-gb level	将数据库进程的优先级设置为 <i>level</i> [Windows、Unix、Mac OS X]。请参见 “-gb 服务器选项” 一节第 178 页。
-gc num	将最大检查点超时周期设置为 <i>num</i> 分钟。请参见 “-gc 服务器选项” 一节第 178 页。
-gd level	设置数据库启动权限。请参见 “-gd 服务器选项” 一节第 179 页。
-ge size	设置运行外部函数的线程的堆栈大小。请参见 “-ge 服务器选项” 一节第 180 页。
-gf	禁止激发触发器。请参见 “-gf 服务器选项” 一节第 180 页。
-gk level	设置停止服务器所需的权限。请参见 “-gk 服务器选项” 一节第 181 页。
-gl level	设置装载或卸载数据所需的权限。请参见 “-gl 服务器选项” 一节第 181 页。
-gm num	设置最大连接数。请参见 “-gm 服务器选项” 一节第 182 页。

服务器选项	说明
-gn num	设置数据库服务器可同时执行的最大任务数。请参见“ -gn 服务器选项 ”一节第 182 页。
-gp size	将最大页面大小设置为 <i>size</i> 字节。请参见“ -gp 服务器选项 ”一节第 183 页。
-gr minutes	设置最长恢复时间。请参见“ -gr 服务器选项 ”一节第 184 页。
-gss size	将线程堆栈大小设置为 <i>size</i> 字节。请参见“ -gss 服务器选项 ”一节第 185 页。
-gt num	设置最多可使用的物理处理器数（不超过授权使用的最大值）。该选项仅在多处理器系统中有用。请参见“ -gt 服务器选项 ”一节第 185 页。
-gtc logical-processors-to-use	控制数据库服务器允许的处理器最大并发数。请参见“ -gtc 服务器选项 ”一节第 186 页。
-gu level	设置实用程序命令的权限级别： <i>utility_db</i> 、 <i>all</i> 、 <i>none</i> 或 <i>DBA</i> 。请参见“ -gu 服务器选项 ”一节第 188 页。
-im submode	在内存中运行数据库服务器，减少或消除到磁盘的写入。请参见“ -im 服务器选项 ”一节第 188 页。
-k	控制性能监控器统计信息的收集。请参见“ -k 服务器选项 ”一节第 190 页。
-kl GSS-API-library-file	指定 Kerberos GSS-API 库（在 Unix 上为共享对象）的文件名称，并允许与数据库服务器建立采用 Kerberos 验证的连接。请参见“ -kl 服务器选项 ”一节第 190 页。
-kr server-realm	指定 Kerberos 服务器主体的域并允许与数据库服务器建立采用 Kerberos 验证的连接。请参见“ -kr 服务器选项 ”一节第 191 页。
-krb	允许与数据库服务器建立采用 Kerberos 验证的连接。请参见“ -krb 服务器选项 ”一节第 192 页。
-ks	禁止创建性能监控器用于从数据库服务器收集计数器值的共享内存 [Windows]。请参见“ -ks 服务器选项 ”一节第 193 页。
-ksc	指定性能监控器可以监控的最大连接数 [Windows]。请参见“ -ksc 服务器选项 ”一节第 193 页。
-ksd	指定性能监控器可以监控的数据库的最大数目 [Windows]。请参见“ -ksd 服务器选项 ”一节第 193 页。

服务器选项	说明
-m	在所有数据库的每个检查点后截断事务日志。请参见“ -m 服务器选项 ”一节第 194 页。
-n name	使用 <i>name</i> 作为数据库服务器的名称。注意，-n 选项与位置有关。请参见“ -n 服务器选项 ”一节第 195 页。
-o filename	将消息输出到指定文件。请参见“ -o 服务器选项 ”一节第 196 页。
-oe filename	指定用于记录启动错误、致命错误和声明的文件。请参见“ -oe 服务器选项 ”一节第 197 页。
-on size	指定数据库服务器消息日志文件的最大大小，达到该大小后，将用扩展名 <i>.old</i> 重命名该文件并起用一个新文件。请参见“ -on 服务器选项 ”一节第 198 页。
-os size	限制消息日志文件的大小。请参见“ -os 服务器选项 ”一节第 198 页。
-ot filename	截断数据库服务器消息日志文件并将输出消息附加到该文件。请参见“ -ot 服务器选项 ”一节第 199 页。
-p packet-size	设置最大的网络包大小 [网络服务器]。请参见“ -p 服务器选项 ”一节第 200 页。
-pc	压缩除在同一计算机的连接之外的所有连接。请参见“ -pc 服务器选项 ”一节第 200 页。
-pt size_in_bytes	设置要压缩的最小网络包大小。请参见“ -pt 服务器选项 ”一节第 201 页。
-qi	不显示数据库服务器系统托盘图标或数据库服务器消息窗口 [Windows]。请参见“ -qi 服务器选项 ”一节第 201 页。
-qn	启动时不最小化数据库服务器消息窗口 [Windows 和 Linux]。请参见“ -qn 服务器选项 ”一节第 202 页。
-qp	取消数据库服务器消息窗口中有关性能的消息。请参见“ -qp 服务器选项 ”一节第 203 页。
-qs	取消启动错误窗口。请参见“ -qs 服务器选项 ”一节第 203 页。
-qw	不显示数据库服务器消息窗口。请参见“ -qw 服务器选项 ”一节第 204 页。
-r	以只读模式打开数据库。请参见“ -r 服务器选项 ”一节第 204 页。

服务器选项	说明
-s <i>facility-ID</i>	设置 Syslog 工具 ID [Unix、Mac OS X]。请参见“ -s 服务器选项 ”一节第 205 页。
-sb { 0 1 }	指定服务器对广播的响应方式。请参见“ -sb 服务器选项 ”一节第 206 页。
-sf <i>feature-list</i>	保护在此数据库服务器上运行的数据库功能的安全。请参见“ -sf 服务器选项 ”一节第 207 页。
-sk <i>key</i>	指定可用于启用对数据库服务器禁用的功能的密钥。请参见“ -sk 服务器选项 ”一节第 210 页。
-su <i>password</i>	设置实用程序数据库 (utility_db) 的 DBA 用户的口令，或禁止与实用程序数据库连接。请参见“ -su 服务器选项 ”一节第 211 页。
-ti <i>minutes</i>	设置关机前的客户端空闲时间—缺省值为 240 分钟。请参见“ -ti 服务器选项 ”一节第 212 页。
-tl <i>seconds</i>	设置缺省的客户端活动超时秒数—缺省值为 120 秒。请参见“ -tl 服务器选项 ”一节第 212 页。
-tmf	强制事务管理器恢复分布式事务 [Windows]。请参见“ -tmf 服务器选项 ”一节第 213 页。
-tmt <i>milliseconds</i>	为分布式事务设置重新征用超时 [Windows]。请参见“ -tmt 服务器选项 ”一节第 214 页。
-tq <i>time</i>	设置退出时间 [网络服务器]。请参见“ -tq 服务器选项 ”一节第 214 页。
-u	使用缓冲磁盘 I/O [Windows、Unix、Mac OS X]。请参见“ -u 服务器选项 ”一节第 215 页。
-ua	禁用异步 I/O [Linux]。请参见“ -ua 服务器选项 ”一节第 215 页。
-uc	以 shell 模式启动数据库服务器 [Unix 和 Mac OS X]。请参见“ -uc 服务器选项 ”一节第 215 页。
-ud	作为守护程序运行 [Unix、Mac OS X]。请参见“ -ud 服务器选项 ”一节第 216 页。
-uf	指定在发生致命错误时要采取的操作 [Unix、Mac OS X]。请参见“ -uf 服务器选项 ”一节第 217 页。
-ui	打开 [服务器启动选项] 窗口并显示数据库服务器消息窗口，或在无可用显示时，以 shell 模式启动数据库服务器 [Linux 和 Mac OS X]。请参见“ -ui 服务器选项 ”一节第 217 页。

服务器选项	说明
-um	打开 [服务器启动选项] 窗口并显示数据库服务器消息窗口 [Mac OS X]。请参见 “-um 服务器选项” 一节第 218 页。
-ut minutes	每 <i>min</i> 分钟处理一次临时文件 [Unix、Mac OS X]。请参见 “-ut 服务器选项” 一节第 218 页。
-ux	显示数据库服务器消息窗口和 [服务器启动选项] 窗口 [Linux]。请参见 “-ux 服务器选项” 一节第 219 页。
-v	显示数据库服务器版本并停止。请参见 “-v 服务器选项” 一节第 220 页。
-vss	启用和禁用卷影复制服务 (Volume Shadow Copy Service, VSS)。请参见 “-vss 服务器选项” 一节第 220 页。
-x list	指定要使用的以逗号分隔的通信链接列表。请参见 “-x 服务器选项” 一节第 221 页。
-xa authentication-info	为仲裁服务器指定数据库名和验证字符串列表。请参见 “-xa 服务器选项” 一节第 222 页。
-xd	防止数据库服务器成为缺省数据库服务器。请参见 “-xd 服务器选项” 一节第 223 页。
-xf state-file	指定用于维护有关数据库镜像系统状态信息的文件的位置。请参见 “-xf 服务器选项” 一节第 223 页。
-xs	指定服务器端 Web 服务通信协议。请参见 “-xs 服务器选项” 一节第 224 页。
-z	提供有关通信链接的诊断信息 [网络服务器]。请参见 “-z 服务器选项” 一节第 226 页。
-ze	在数据库服务器消息窗口中显示数据库服务器环境变量。请参见 “-ze 服务器选项” 一节第 226 页。
-zl	为各个连接启用最新预准备的 SQL 语句的捕获功能。请参见 “-zl 服务器选项” 一节第 227 页。
-zn integer	指定要保留的请求日志文件副本的数量。请参见 “-zn 服务器选项” 一节第 228 页。
-zo filename	将请求记录信息重定向到单独的文件。请参见 “-zo 服务器选项” 一节第 229 页。

服务器选项	说明
-zoc	将 Web 服务客户端信息重定向到文件。请参见“-zoc 服务器选项”一节第 229 页。
-zp	启用捕获查询优化程序最近使用的计划。请参见“-zp 服务器选项”一节第 230 页。
-zr { all SQL none }	启用 SQL 操作的记录。缺省为 NONE。请参见“-zr 服务器选项”一节第 230 页。
-zs size	限制用于记录请求的日志文件的大小。请参见“-zs 服务器选项”一节第 232 页。
-zt	启用对请求计时信息的记录。请参见“-zt 服务器选项”一节第 233 页。

数据库选项

以下选项只能在数据库服务器命令中的数据库文件名之后指定。

数据库选项	说明
-a filename	应用指定的事务日志文件。请参见“-a 数据库选项”一节第 234 页。
-ad log-directory	指定包含要应用于数据库的事务日志文件的目录。请参见“-ad 数据库选项”一节第 234 页。
-ar	将任何与事务日志位于相同目录的日志文件应用于数据库。请参见“-ar 数据库选项”一节第 235 页。
-as	在应用事务日志后继续运行数据库（与 -ad 或 -ar 一起使用）。请参见“-as 数据库选项”一节第 236 页。
-dh	在对此服务器使用 dblocate 时不显示数据库。请参见“-dh 数据库选项”一节第 237 页。
-ds	指定数据库 dbspace 的位置。请参见“-ds 数据库选项”一节第 237 页。
-ek key	指定加密密钥。请参见“-ek 数据库选项”一节第 238 页。
-m	在指定数据库的每个检查点之后截断（删除）事务日志。请参见“-m 数据库选项”一节第 238 页。
-n name	为数据库命名。请参见“-n 数据库选项”一节第 239 页。
-r	以只读模式打开指定的数据库。不允许修改数据库。请参见“-r 数据库选项”一节第 240 页。

数据库选项	说明
-sm	提供可用于访问只读镜像数据库的数据库服务器名称。请参见“ -sm 数据库选项 ”一节第 241 页。
-sn alternate-server-name	为运行于数据库服务器上的单个数据库提供替代服务器名。请参见“ -sn 数据库选项 ”一节第 242 页。
-xp mirroring-options	使用数据库镜像时，为运行服务器提供信息，使其能与伙伴服务器和仲裁服务器相连。请参见“ -xp 数据库选项 ”一节第 244 页。

注释

dbeng11 命令用于启动个人数据库服务器。**dbsrv11** 命令用于启动网络数据库服务器。

database-file 用于指定数据库文件名。如果指定的 *database-file* 没有文件扩展名，则 SQL Anywhere 将查找扩展名为 *.db* 的 *database-file*。如果使用相对路径，则相对当前工作目录获取路径。您可以提供完整路径。

如果想要从批处理文件启动数据库服务器，则必须使用 **dbspawn** 实用程序。请参见“[后台启动服务器实用程序 \(dbspawn\)](#)”一节第 781 页。

个人数据库服务器的并发连接数最大值为 10 个，最多使用一个 CPU 进行请求处理，而且不支持网络客户端/服务器连接。

此外，还有其它微小差异，例如，启动新数据库所需的缺省权限级别，或执行 CHECKPOINT 语句所需的权限。

为每个支持的操作系统同时提供个人和网络数据库服务器，下面的情况例外。在 Windows Mobile 上，只提供网络服务器。在网络服务器中支持 TCP/IP，使您可以从台式计算机上使用 Sybase Central 执行任务，包括进行数据库管理。

示例

以下命令用于启动在个人数据库服务器上运行的 SQL Anywhere 示例数据库：

```
dbeng11 "c:\Documents and Settings\All Users\Documents\SQL Anywhere 11\Samples\demo.db"
```

以下命令用于启动在网络数据库服务器上运行的 SQL Anywhere 示例数据库：

```
dbsrv11 "c:\Documents and Settings\All Users\Documents\SQL Anywhere 11\Samples\demo.db"
```

在以下示例中（所有内容均在一行中输入）启动一个名为 **myserver** 的服务器，该服务器以 3 MB 大小的高速缓存启动，并装载示例数据库：

```
dbeng11 -c 3m -n myserver "samples-dir\demo.db"
```

有关 *samples-dir* 的信息，请参见“[示例目录](#)”一节第 368 页。

数据库服务器选项

这些选项适用于整个服务器，而不是只适用于个别数据库。

@data 服务器选项

从指定的环境变量或配置文件读入选项。

语法

```
{ dbsrv11 | dbeng11 } @data ...
```

适用于

所有操作系统和数据库服务器，Windows Mobile 除外。除语言选择实用程序 (dblang)、重建实用程序 (rebuild)、证书创建实用程序 (createcert)、证书查看器实用程序 (viewcert)、ActiveSync 提供程序安装实用程序 (mlasinst) 和文件隐藏实用程序 (dbfhide) 之外的所有数据库实用程序都支持这些选项。

注释

使用此选项从指定的环境变量或配置文件中读入命令行选项。如果同时存在名称相同的环境变量和配置文件，则使用环境变量。

配置文件可以包含换行符，并且可以包含任何选项集。请参见“使用配置文件”一节第 695 页。

如果要保护配置文件中的信息（例如该文件包含口令），可以使用文件隐藏 (dbfhide) 实用程序对配置文件的内容进行模糊处理。请参见“文件隐藏实用程序 (dbfhide)”一节第 724 页。

@data 参数可以出现在命令行中的任何位置，文件中所含的参数会插入到该位置。可以指定多个文件，并且文件指定符可以与命令行选项一起使用。

另请参见

- “使用配置文件”一节第 695 页

示例

以下配置文件包含一组选项，这些选项适用于一个名为 **myserver** 的服务器，它以 4 MB 大小的高速缓存启动，并装载示例数据库：

```
-c 4096
-n myservers
"c:\mydatabase.db"
```

如果此配置文件保存为 `c:\config.txt`，则在命令中可按如下方式使用：

```
dbsrv11 @c:\config.txt
```

以下配置文件包含注释：

```
#This is the server name:
-n MyServer
#These are the protocols:
-x tcpip
```

```
#This is the database file  
my.db
```

以下语句将设置一个包含数据库服务器选项的环境变量，该数据库服务器以 4 MB 大小的高速缓存启动，并装载示例数据库。

```
SET envvar=-c 4096 "c:\mydatabase.db";
```

此命令使用名为 **envvar** 的环境变量启动数据库服务器。

```
dbsrv11 @envvar
```

-? 服务器选项

显示用法信息。

语法

```
{ dbsrv11 | dbeng11 } -?
```

适用于

所有操作系统和数据库服务器，Windows Mobile 除外。

注释

显示各个服务器选项的简短说明。数据库不执行任何其它任务。

-b 服务器选项

使用批量操作模式。

语法

```
{ dbsrv11 | dbeng11 } -b ...
```

适用于

所有操作系统和数据库服务器。

注释

它对使用 Interactive SQL INPUT 命令将大量数据装载到数据库非常有用。

如果正在使用 LOAD TABLE 批量装载数据，则不应使用 **-b** 选项。

当使用此选项时，数据库服务器只允许一个应用程序使用一个连接。它会保留回退日志，但不保留事务日志。它将关闭多用户锁定机制。

在使用 **-b** 选项装载数据后首次启动数据库服务器时，应使用新日志文件。

批量操作模式不禁用触发器的触发。

另请参见

- “批量操作的数据恢复问题”一节 《SQL Anywhere 服务器 - SQL 的用法》
- “批量操作的性能问题”一节 《SQL Anywhere 服务器 - SQL 的用法》

-c 服务器选项

设置为高速缓存数据库页面和其它服务器信息而保留的初始内存。

语法

```
{ dbsrv11 | dbeng11 } -c { size[ k | m | g | p ] } ...
```

适用于

所有操作系统和数据库服务器。

注释

可用作数据库服务器高速缓存的内存量是控制性能的关键因素之一。可以使用 -c 服务器选项设置高速缓存的初始值。提供给服务器的高速缓存越大，性能就越高。

size 是以字节为单位的内存量。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。

单位 **p** 是物理系统内存的百分比或非 AWE 高速缓存最大值的百分比，取其低者。非 AWE 高速缓存最大值取决于操作系统。例如：

- Windows 32 位 Advanced Server、Enterprise Server、Datacenter Server 和 Vista 为 2.8 GB
- 运行于 Windows x64 Edition 上的 32 位数据库服务器为 3.8 GB
- 其它所有 32 位系统上均为 1.8 GB
- 在 Windows Mobile 上，**p** 选项指定可用物理内存的百分比

如果使用 **p**，则此参数为一个百分比。您可以使用 % 来代替 **p**，但由于大多数非 Unix 操作系统使用 % 作为环境变量转义字符，因此必须将 % 字符转义。要将初始高速缓存大小设置为物理系统内存的 50%，可使用以下方法：

```
dbeng11 -c 50%% ...
```

在 Unix 操作系统上，高速缓存的大小将设置为下列各项中的较低者：

- -c 后指定的值
- (*available memory* - 5 MB) 的 95%

在 Windows Mobile 上，高速缓存大小将设置为下列各项中的较小者：

- -c 后指定的值
- (*available memory* - 2 MB) 的 95%

如果没有提供 -c 选项，数据库服务器将按下列方式计算初始高速缓存分配：

1. 它使用以下操作系统特定的缺省高速缓存大小：
 - **Windows Mobile** 600 KB
 - **Windows** 2 MB
 - **Unix** 8 MB
2. 它计算运行时特定的最小缺省高速缓存大小，该数值为下列各项中较小者：
 - 计算机物理内存的 25%
 - 命令行中指定的主数据库文件大小的总和。计算时不包括主数据库文件以外的其它 `dbspace`。如果没有指定文件，则该值为零。
3. 分配两个计算值中的较大者。

如果数据库已加密，您最好增加高速缓存的大小。此外，如果您要使用动态调整高速缓存大小的功能（`-ca` 选项），所用高速缓存大小可能会受可用内存量的限制。

请参见“增加高速缓存大小”一节《SQL Anywhere 服务器 - SQL 的用法》。

数据库服务器消息窗口显示启动时的高速缓存大小，可以使用以下语句来获得当前高速缓存的大小：

```
SELECT PROPERTY( 'CacheSize' );
```

另请参见

- “`-ca` 服务器选项”一节第 159 页
- “`-cc` 服务器选项”一节第 160 页
- “`-ch` 服务器选项”一节第 161 页
- “`-cl` 服务器选项”一节第 162 页
- “`-cm` 服务器选项”一节第 163 页
- “`-cr` 服务器选项”一节第 165 页
- “`-cs` 服务器选项”一节第 165 页
- “`-cv` 服务器选项”一节第 166 页
- “`-cw` 服务器选项”一节第 167 页
- “限制高速缓存内存使用”一节《SQL Anywhere 服务器 - SQL 的用法》
- “增加高速缓存大小”一节《SQL Anywhere 服务器 - SQL 的用法》
- “使用高速缓存提高性能”一节《SQL Anywhere 服务器 - SQL 的用法》

示例

在以下示例中（所有内容均在一行中输入）启动一个名为 `myserver` 的服务器，该服务器以 3 MB 大小的高速缓存启动，并装载示例数据库：

```
dbeng11 -c 3m -n myservers "samples-dir\demo.db"
```

有关 `samples-dir` 的信息，请参见“示例目录”一节第 368 页。

-ca 服务器选项

强制实施静态高速缓存大小。

语法

```
{ dbsrv11 | dbeng11 } -ca 0 ...
```

适用于

Windows、Unix、Mac OS X

注释

可以通过在命令行指定 `-ca 0` 来禁止因服务器负载高而自动增加高速缓存。如果不包括 `-ca 0` 选项，数据库服务器则会自动增加高速缓存大小。高速缓存的不断增大可能会导致数据库服务器遇到 [致命错误：动态内存已耗尽] 错误。

使用此服务器选项时必须只采用 `-ca 0` 的格式。

如果使用 AWE 高速缓存，则忽略此选项。可以使用 `-cw` 选项创建更大的使用 AWE 的高速缓存。请参见“[-cw 服务器选项](#)”一节第 167 页。

另请参见

- “[-c 服务器选项](#)”一节第 158 页
- “[-cc 服务器选项](#)”一节第 160 页
- “[-ch 服务器选项](#)”一节第 161 页
- “[-cl 服务器选项](#)”一节第 162 页
- “[-cm 服务器选项](#)”一节第 163 页
- “[-cr 服务器选项](#)”一节第 165 页
- “[-cs 服务器选项](#)”一节第 165 页
- “[-cv 服务器选项](#)”一节第 166 页
- “[-cw 服务器选项](#)”一节第 167 页
- “[限制高速缓存内存使用](#)”一节 《[SQL Anywhere 服务器 - SQL 的用法](#)》

示例

以下示例启动一个名为 **myserver** 的服务器，其静态高速缓存为可用物理内存的 40%，并装载示例数据库：

```
dbsrv11 -c 40P -ca 0 -n myservers "samples-dir\demo.db"
```

有关 *samples-dir* 的信息，请参见“[示例目录](#)”一节第 368 页。

-cc 服务器选项

收集有关下次启动数据库时用于缓存预热的数据库页的信息。

语法

```
{ dbsrv11 | dbeng11 } -cc { + | - } ...
```

适用于

所有操作系统和数据库服务器。

注释

缺省情况下，将启用页收集。启用收集后，数据库服务器将跟踪所请求的每个数据库页。出现以下情况时将停止收集：收集的页数达到最大值，数据库关闭，或者收集率降至最小值以下。注意，不能配置收集的最大页数，也不能指定收集率的值（该值基于高速缓存大小和数据库大小）。收集停止后，将在数据库中记录有关请求页的信息，这样就可以在下次用 `-cr` 选项启动数据库时使用这些页来预热缓存。缺省情况下，将启用引用页的收集。

另请参见

- “`-c` 服务器选项” 一节第 158 页
- “`-ca` 服务器选项” 一节第 159 页
- “`-ch` 服务器选项” 一节第 161 页
- “`-cl` 服务器选项” 一节第 162 页
- “`-cm` 服务器选项” 一节第 163 页
- “`-cr` 服务器选项” 一节第 165 页
- “`-cs` 服务器选项” 一节第 165 页
- “`-cv` 服务器选项” 一节第 166 页
- “`-cw` 服务器选项” 一节第 167 页
- “使用高速缓存预热” 一节 《SQL Anywhere 服务器 - SQL 的用法》

-ch 服务器选项

设置最大高速缓存大小，作为对高速缓存自动增长的限制。

语法

```
{ dbsrv11 | dbeng11 } -ch { size[ k | m | g | p ] } ...
```

适用于

Windows、Unix、Mac OS X

注释

此选项限制高速缓存自动增长的过程中数据库服务器高速缓存的大小。缺省情况下，上限大约为非 AWE 高速缓存最大值和计算机物理内存的 90% 这二者中的较低者。

size 是以字节为单位的内存量。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。

单位 **p** 是物理系统内存的百分比或非 AWE 高速缓存最大值的百分比，取其低者。非 AWE 高速缓存最大值取决于操作系统。例如：

- Windows 32 位 Advanced Server、Enterprise Server 和 Datacenter Server 为 2.8 GB
- 运行于 Windows x64 Edition 上的 32 位数据库服务器为 3.8 GB
- 其它所有 32 位系统上均为 1.8 GB
- 在 Windows Mobile 上，**p** 选项指定可用物理内存的百分比

如果使用 **p**，则此参数为一个百分比。您可以使用 % 来代替 **P**，但由于大多数非 Unix 操作系统使用 % 作为环境变量转义字符，因此必须将 % 字符转义。要将最小高速缓存大小设置为物理系统内存的 50%，可使用以下方法：

如果使用 AWE 高速缓存，则忽略此选项。可以使用 **-cw** 选项创建更大的使用 AWE 的高速缓存。请参见“**-cw 服务器选项**”一节第 167 页。

```
dbeng11 -ch 50%% ...
```

另请参见

- “**-c 服务器选项**”一节第 158 页
- “**-ca 服务器选项**”一节第 159 页
- “**-cc 服务器选项**”一节第 160 页
- “**-cl 服务器选项**”一节第 162 页
- “**-cm 服务器选项**”一节第 163 页
- “**-cr 服务器选项**”一节第 165 页
- “**-cs 服务器选项**”一节第 165 页
- “**-cv 服务器选项**”一节第 166 页
- “**限制高速缓存内存使用**”一节 《SQL Anywhere 服务器 - SQL 的用法》

示例

以下示例启动一个名为 **silver** 的服务器，其最大高速缓存大小为 2 MB，并装载示例数据库：

```
dbeng11 -ch 2m -n silver "samples-dir\demo.db"
```

有关 *samples-dir* 的信息，请参见“**示例目录**”一节第 368 页。

-cl 服务器选项

设置最小高速缓存大小，将其做为高速缓存大小自动调整的下限。

语法

```
{ dbsrv11 | dbeng11 } -cl { size [ k | m | g | p ] } ...
```

适用于

Windows、Unix、Mac OS X

注释

此选项设置高速缓存的下限。如果使用 **-c** 选项指定了初始高速缓存大小，则最小高速缓存大小与初始高速缓存大小相同。如果未指定初始高速缓存大小，则缺省初始高速缓存大小在 Windows 上为 2 MB，在 Unix 上为 8 MB。

size 是以字节为单位的内存量。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。

单位 **p** 是物理系统内存的百分比或非 AWE 高速缓存最大值的百分比，取其低者。非 AWE 高速缓存最大值取决于操作系统。例如：

- Windows 32 位 Advanced Server、Enterprise Server 和 Datacenter Server 为 2.8 GB

- 运行于 Windows x64 Edition 上的 32 位数据库服务器为 3.8 GB
- 其它所有 32 位系统上均为 1.8 GB
- 在 Windows Mobile 上，**p** 选项指定可用物理内存的百分比

如果使用 **p**，则此参数为一个百分比。您可以使用 % 来代替 **P**，但由于大多数非 Unix 操作系统使用 % 作为环境变量转义字符，因此必须将 % 字符转义。要将最小高速缓存大小设置为物理系统内存的 50%，可使用以下方法：

```
dbeng11 -cl 50%% ...
```

如果使用 AWE 高速缓存，则忽略此选项。可以使用 **-cw** 选项创建更大的使用 AWE 的高速缓存。请参见“**-cw** 服务器选项”一节第 167 页。

另请参见

- “**-c** 服务器选项”一节第 158 页
- “**-ca** 服务器选项”一节第 159 页
- “**-cc** 服务器选项”一节第 160 页
- “**-ch** 服务器选项”一节第 161 页
- “**-cm** 服务器选项”一节第 163 页
- “**-cr** 服务器选项”一节第 165 页
- “**-cs** 服务器选项”一节第 165 页
- “**-cv** 服务器选项”一节第 166 页
- “**-cw** 服务器选项”一节第 167 页
- “限制高速缓存内存使用”一节 《SQL Anywhere 服务器 - SQL 的用法》

示例

以下示例启动一个名为 silver 的服务器，其最小高速缓存大小为 5 MB，并装载数据库文件 *example.db*：

```
dbeng11 -cl 5m -n silver "c:\example.db"
```

-cm 服务器选项

指定为 Windows 上的 Address Windowing Extensions (AWE) 高速缓存分配的地址空间量。

语法

```
{ dbsrv11 | dbeng11 } -cm { size[ k | m | g | p ] } ...
```

适用于

Windows

注释

在任何支持的平台上使用 AWE 高速缓存时，数据库服务器会将除 512 MB 之外的整个地址空间用来访问高速缓存。这 512 MB 地址空间留作其它用途，例如数据库必须装载的 DLL 或用于非高速缓存的内存分配。对于大多数系统，缺省设置就已足够。如果您需要增加或减少保留的地址空间

量，可通过指定 `-cm` 选项来实现。数据库服务器在启动时会在数据库服务器消息窗口中显示其使用的地址空间量。

`size` 是以字节为单位的内存量。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。

单位 **p** 为非 AWE 高速缓存最大值的百分比。如果使用 **p**，则此参数为一个百分比。您可以使用 `%` 来代替 **P**，但由于大多数非 Unix 操作系统使用 `%` 作为环境变量转义字符，因此必须将 `%` 字符转义。要将高速缓存大小设置为地址空间的 50%，可使用以下方法：

```
dbeng11 -cm 50%% ...
```

另请参见

- “[-c 服务器选项](#)” 一节第 158 页
- “[-ca 服务器选项](#)” 一节第 159 页
- “[-cc 服务器选项](#)” 一节第 160 页
- “[-ch 服务器选项](#)” 一节第 161 页
- “[-cl 服务器选项](#)” 一节第 162 页
- “[-cr 服务器选项](#)” 一节第 165 页
- “[-cs 服务器选项](#)” 一节第 165 页
- “[-cv 服务器选项](#)” 一节第 166 页
- “[-cw 服务器选项](#)” 一节第 167 页
- “[限制高速缓存内存使用](#)” 一节 《[SQL Anywhere 服务器 - SQL 的用法](#)》

-cp 服务器选项

指定在其中搜索类的一组目录或 jar 文件。

语法

```
{ dbsrv11 | dbeng11 } -cp location[ ;location ... ] ...
```

适用于

所有操作系统和数据库服务器。

注释

建议将要与数据库中的 Java 一起使用的所有类和 JAR 文件都安装在数据库中。如果将这些类和 JAR 文件存储在数据库中，则能够将该数据库轻松地移动到其它计算机或操作系统上。将这些类和 JAR 文件安装在数据库中的另一个好处是，SQL Anywhere 类装载程序能够从数据库中获取类和资源，这样，使用数据库中的 Java 的每个连接都可以有自己的关于这些类的实例，以及在这些类中有自己的静态变量副本。

但在必须由系统类装载程序来装载类或 JAR 文件的情况下，可通过 `-cp` 服务器选项来指定它们。此选项用于将目录和 JAR 文件添加到数据库服务器所构建的用于启动 Java VM 的类路径中。

另请参见

- “[Java 支持简介](#)” 一节 《[SQL Anywhere 服务器 - 编程](#)》
- “[如何在数据库中存储 Java 类？](#)” 一节 《[SQL Anywhere 服务器 - 编程](#)》

-cr 服务器选项

使用上次运行数据库时所收集的信息、以数据库页重装（预热）高速缓存。

语法

```
{ dbsrv11 | dbeng11 } -cr { + | - } ...
```

适用于

所有操作系统和数据库服务器。

注释

可以指示数据库服务器用上次启动数据库时所引用的页来预热高速缓存（使用 `-cc` 选项启用页收集）。缺省情况下，将启用高速缓存预热。数据库启动后，服务器将对其进行检查，看它是否包含上次启动数据库时所请求的页的集合的信息。如果数据库包含此信息，随后即会将先前引用的页装载到高速缓存中。

如果在每次启动数据库后要对它执行的查询相同或相似，那么用上次启动数据库时所引用的页来预热高速缓存可以提高性能。

另请参见

- [“-cc 服务器选项”一节第 160 页](#)
- [“-cl 服务器选项”一节第 162 页](#)
- [“-cm 服务器选项”一节第 163 页](#)
- [“-cs 服务器选项”一节第 165 页](#)
- [“-cv 服务器选项”一节第 166 页](#)
- [“-cw 服务器选项”一节第 167 页](#)
- [“使用高速缓存预热”一节 《SQL Anywhere 服务器 - SQL 的用法》](#)

-cs 服务器选项

在数据库服务器消息窗口中显示高速缓存大小的变化。

语法

```
{ dbsrv11 | dbeng11 } -cs ...
```

适用于

Windows、Unix

注释

出于排除故障的目的，每当高速缓存大小发生变化时，数据库服务器消息窗口中就会显示高速缓存信息。

另请参见

- “-c 服务器选项” 一节第 158 页
- “-ca 服务器选项” 一节第 159 页
- “-cc 服务器选项” 一节第 160 页
- “-ch 服务器选项” 一节第 161 页
- “-cl 服务器选项” 一节第 162 页
- “-cm 服务器选项” 一节第 163 页
- “-cr 服务器选项” 一节第 165 页
- “-cv 服务器选项” 一节第 166 页
- “-cw 服务器选项” 一节第 167 页
- “使用高速缓存预热” 一节 《SQL Anywhere 服务器 - SQL 的用法》

-cv 服务器选项

控制是否在数据库服务器消息窗口中显示有关高速缓存预热的消息。

语法

```
{ dbsrv11 | dbeng11 } -cv { + | - } ...
```

适用于

所有操作系统和数据库服务器。

注释

指定 -cv+ 后，如果出现下列任意一种高速缓存预热活动，就会在数据库服务器消息窗口中显示一条消息：

- 开始或停止收集请求的页（由 -cc 服务器选项控制）
- 开始或停止重装页（由 -cr 服务器选项控制）

缺省情况下，此选项是关闭的。

另请参见

- “-c 服务器选项” 一节第 158 页
- “-ca 服务器选项” 一节第 159 页
- “-cc 服务器选项” 一节第 160 页
- “-ch 服务器选项” 一节第 161 页
- “-cl 服务器选项” 一节第 162 页
- “-cm 服务器选项” 一节第 163 页
- “-cr 服务器选项” 一节第 165 页
- “-cs 服务器选项” 一节第 165 页
- “-cw 服务器选项” 一节第 167 页
- “使用高速缓存预热” 一节 《SQL Anywhere 服务器 - SQL 的用法》

示例

以下命令将启动数据库 *mydatabase.db*（开启数据库页收集和页装载功能），并将有关这些活动的消息记录到数据库服务器消息窗口中：

```
dbsrv11 -cc+ -cr+ -cv+ mydatabase.db
```

-cw 服务器选项

在 Windows 中允许使用 Address Windowing Extensions (AWE) 来设置数据库服务器高速缓存的大小。

语法

```
{ dbsrv11 | dbeng11 } -cw ...
```

适用于

Windows

注释

可用作数据库服务器高速缓存的内存量是控制性能的关键因素之一。由于 Windows 支持 Address Windowing Extensions，因此您可以使用 -cw 选项来利用基于系统中最大物理内存量的大容量高速缓存。

在 64 位的 SQL Anywhere 数据库服务器上不支持 AWE 高速缓存。

操作系统	非 AWE 高速缓存最大值	Windows 支持的最大物理内存
Windows 2000 Professional	1.8 GB	4 GB
Windows 2000 Server	1.8 GB	4 GB
Windows 2000 Advanced Server	2.7 GB ¹	8 GB
Windows 2000 Datacenter Server	2.7 GB ¹	64 GB
Windows XP Home Edition	1.8 GB	2 GB
Windows XP Professional	1.8 GB	4 GB
Windows Server 2003, Web Edition	1.8 GB	2 GB
Windows Server 2003, Standard Edition	1.8 GB	4 GB
Windows Server 2003, Enterprise Edition	2.7 GB ¹	32 GB

操作系统	非 AWE 高速缓存最大值	Windows 支持的最大物理内存
Windows Server 2003, Datacenter Edition	2.7 GB ¹	64 GB
Windows Vista Ultimate	2.7 GB ²	4 GB
Windows Vista Enterprise	2.7 GB ²	4 GB
Windows Vista Business	2.7 GB ²	4 GB
Windows Vista Home Premium	2.7 GB ²	4 GB
Windows Vista Home Basic	2.7 GB ²	4 GB
Windows Vista Starter	2.7 GB ²	1 GB

¹ 在 Windows XP/200x 上，必须使用 **/3GB** 选项来启动操作系统，这样才能使用这一容量的高速缓存。

² 在 Windows Vista 上，必须先以管理员身份运行以下命令重新启动操作系统，这样才能使用这一容量的高速缓存：

```
bcdedit /set increaseuserva 3072
```

使用 AWE 高速缓存时，系统中大多数可用物理内存都可分配给高速缓存。

建议尽可能以非 AWE 高速缓存来设置所需大小的高速缓存，因为 AWE 高速缓存分配的内存只能供数据库服务器使用。这意味着在数据库服务器运行时，操作系统和其它应用程序不能使用为数据库服务器高速缓存分配的内存。AWE 高速缓存不支持动态高速缓存调整。因此，如果使用 AWE 高速缓存，并且指定 **-ch** 或 **-cl** 选项来设置高速缓存大小的上限和下限，这些设置将被忽略。

缺省情况下，保留 512 MB 的地址空间用于 SQL Anywhere AWE 高速缓存之外的其它用途（地址空间是指可由程序在任意指定时间访问的内存量）。尽管此缺省空间量在多数情况下已足够，但可以使用 **-cm** 选项更改保留的地址空间量。

在 Windows Vista 上，只有提升的数据库服务器才能使用 AWE 内存。如果在 Windows Vista 上自动启动数据库服务器，则必须在连接字符串中指定 **ELEVATE=YES**，以便提升自动启动的数据库服务器的可执行能力。请参见“[Elevate 连接参数](#)”一节第 264 页。

要使用 AWE 高速缓存启动数据库服务器，必须执行以下步骤：

- 在 Windows Vista 上，必须以管理员身份运行数据库服务器。
- 系统中至少有 130 MB 的可用内存。
- 在 Windows XP/200x 上，如果系统内存介于 2 GB 和 16 GB 之间，则将 **/3GB** 选项添加到 *boot.ini* 文件 "[operating systems]" 区段中的 Windows 引导行。

在 Windows Vista 上，如果系统内存介于 2 GB 和 16 GB 之间，则必须以管理员身份运行以下命令：

```
bcdedit /set increaseuserva 3072
```

在 Windows XP/200x 上，如果系统内存大于 16 GB，则不要将 **/3GB** 选项添加到 *boot.ini* 文件 [operating systems] 区段中的 Windows 引导行，因为 Windows 无法对超出 16 GB 的内存进行寻址。

- 在 Windows XP/200x 上，如果系统内存大于 4 GB，可将 **/PAE** 选项添加到 *boot.ini* 文件 [operating systems] 区段中的 Windows 引导行。

在 Windows Vista 上，如果系统内存大于 4 GB，则以管理员身份运行以下命令：

```
bcdedit /set pae ForceEnable
```

- 为运行服务器所用的用户 ID 授予 [内存中锁定页] 特权。以下步骤说明如何在 Windows XP 中执行此操作。

1. 以管理员身份登录 Windows。
2. 打开 [控制面板]。
3. 双击 [管理工具]。
4. 双击 [本地安全策略]。
5. 打开左窗格中的 [本地策略]。
6. 双击 [用户权利指派]。
7. 双击右窗格中的 [内存中锁定页] 策略。
8. 单击 [添加用户或组]。
9. 键入用户的名称，然后单击 [确定]。
10. 在 [内存中锁定页] 窗口上单击 [确定]。
11. 关闭所有打开的窗口，然后重新启动计算机以使设置生效。

如果在命令行中指定 **-cw** 选项和 **-c** 选项，则数据库服务器将尝试按以下步骤分配初始高速缓存：

1. AWE 高速缓存不大于 **-c** 选项指定的高速缓存大小。如果 **-c** 选项指定的值小于 2 MB，则不使用 AWE。
2. AWE 高速缓存不大于所有可用物理内存减去 128 MB 的差。
3. AWE 高速缓存不小于 2 MB。如果无法使用此最小物理内存量，则不使用 AWE 高速缓存。

当指定 **-cw** 选项且不指定 **-c** 选项时，数据库服务器将尝试按以下步骤分配初始高速缓存：

1. 除保留 128 MB 内存用于操作系统外，AWE 高速缓存可 100% 使用剩余的全部可用内存。
2. AWE 高速缓存不大于在命令行中指定的主数据库文件大小的总和。计算时不包括主数据库文件以外的其它 *dbspace*。如果没有指定文件，则该值为零。
3. AWE 高速缓存不小于 2 MB。如果无法使用此最小物理内存量，则不使用 AWE 高速缓存。

当服务器使用 AWE 高速缓存时，高速缓存页面大小至少为 4KB，且禁用动态高速缓存调整。

请参见“使用高速缓存提高性能”一节《SQL Anywhere 服务器 - SQL 的用法》。

另请参见

- “-c 服务器选项” 一节第 158 页
- “-ca 服务器选项” 一节第 159 页
- “-cc 服务器选项” 一节第 160 页
- “-ch 服务器选项” 一节第 161 页
- “-cl 服务器选项” 一节第 162 页
- “-cm 服务器选项” 一节第 163 页
- “-cr 服务器选项” 一节第 165 页
- “-cs 服务器选项” 一节第 165 页
- “-cv 服务器选项” 一节第 166 页

示例

以下示例启动一个名为 **myserver** 的服务器，它以 12 GB 高速缓存启动，并装载数据库 `c:\test\mydemo.db`：

```
dbeng11 -n myservers -c 12G -cw c:\test\mydemo.db
```

-dt 服务器选项

指定临时文件的存储目录。

语法

```
{ dbsrv11 | dbeng11 } -dt temp-file-dir ...
```

适用于

所有服务器和操作系统（Unix 上的共享内存连接除外）。

注释

SQL Anywhere 创建了两类临时文件：与数据库服务器有关的临时文件（在所有平台均创建）和与通信有关的临时文件（只在 Unix 上为客户端和服务端创建）。

可使用 `-dt` 选项为与数据库服务器有关的临时文件指定目录。如果在启动数据库服务器时不指定该选项，则 SQL Anywhere 会按显示的顺序检查以下环境变量，以确定将临时文件放置在哪个目录中。

- SATMP
- TMP
- TMPDIR
- TEMP

如果这些环境变量均未定义，则在 Windows 中，SQL Anywhere 会将其临时文件放在当前目录中，而在 Unix 中，则会放在 `/tmp` 目录中。

Unix 上用于通信的临时文件不放在 `-dt` 所指定的目录中。而是对环境变量进行检查，并使用 `/tmp`（如果未定义任何环境变量）。

另请参见

- “数据库文件概述”一节第 12 页
- “SATMP 环境变量”一节第 357 页
- “将不同的文件放置在不同的设备上”一节 《SQL Anywhere 服务器 - SQL 的用法》
- “sa_disk_free_space 系统过程”一节 《SQL Anywhere 服务器 - SQL 参考》
- “temp_space_limit_check 选项 [数据库]”一节第 552 页

-ec 服务器选项

使用传送层安全或简单加密对往来于所有客户端的所有本地 SQL Anywhere 包（DBLib、ODBC 和 OLE DB）进行加密。不加密 TDS 包。

语法

```
{ dbsrv11 | dbeng11 } -ec encryption-options ...
```

encryption-options :

```
{ NONE |
  SIMPLE |
  TLS ( TLS_TYPE=cipher;
    [ FIPS={ Y | N }; ]
    IDENTITY=server-identity-filename;
    IDENTITY_PASSWORD=password ) }, ...
```

适用于

NONE 和 SIMPLE 适用于所有服务器和操作系统。

TLS 适用于所有服务器和操作系统，Windows Mobile 除外。

有关 FIPS 支持的信息，请参见 <http://www.sybase.com/detail?id=1062623>。

注释

您可以使用此选项，利用传送层安全功能来保护客户端应用程序和数据库服务器之间传输的通信包的安全。请参见“[传送层安全](#)”第 1031 页。

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》。

-ec 选项指示数据库服务器只接受使用一种指定类型进行加密的连接。无论是否使用 -ec 选项，始终接受通过 TDS 协议建立的连接（包括使用 jConnect 的 Java 应用程序），并从不对其进行加密。如果将 TDS 协议选项设置为 NO，则不允许建立这些未加密的 TDS 连接。请参见“[TDS 协议选项](#)”一节第 305 页。

缺省情况下，通信包是不加密的，但这样可能会引起潜在的安全风险。如果您注重网络包的安全性，可使用 -ec 选项。加密对于性能只有很轻微的影响。-ec 选项控制服务器的加密设置，并至少需要在逗号分隔的列表中有以下参数之一：

- **NONE** 接受未加密的连接。
- **SIMPLE** 接受使用简单加密技术加密的连接。所有平台以及以前版本的 SQL Anywhere 都支持此类加密。简单加密不提供服务器验证、高度椭圆曲线加密、RSA 加密或其它传送层安全功能。
- **TLS** 接受加密的连接。TLS 参数接受以下必需的参数：

- **cipher** 对于 RSA 和 ECC 加密，可分别为 **RSA** 或 **ECC**。对于 FIPS 认可的 RSA 加密，请指定 **TLS_TYPE=RSA;FIPS=Y**。RSA FIPS 使用单独的认可库，但是与使用 SQL Anywhere 9.0.2 或更高版本指定 RSA 的客户端兼容。

有关支持 FIPS 的平台的列表，请参见 <http://www.sybase.com/detail?id=1062623>。

编码器必须与用于创建证书的加密（ECC 或 RSA）匹配。

有关实施经 FIPS 认可的算法的信息，请参见“-fips 服务器选项”一节第 176 页。

注意

版本 10 及更高版本的客户端无法使用 ECC 算法与版本 9.0.2 或更早版本的数据库服务器建立连接。如果要对此配置进行高度加密，请使用 RSA 算法。

- **server-identity-filename** 是服务器身份证书的路径和文件名。如果使用 FIPS 认可的 RSA 加密，必须使用 RSA 编码器生成证书。

有关创建可以自签名或由证书颁发机构或企业根证书签名的服务器证书的详细信息，请参见“创建数字证书”一节第 1036 页。

- **password** 是服务器专用密钥的口令。在创建服务器证书时指定此口令。

如果数据库服务器接受简单加密，但不接受未加密的连接，则所有未使用加密的非 TDS 连接尝试会自动使用简单加密。

使用 **-ec SIMPLE** 启动数据库服务器将通知数据库服务器仅接受使用简单加密的连接。TLS 连接（ECC、RSA 和 RSA FIPS）将失败，无加密要求的连接会使用简单加密。

使用 **-ec SIMPLE, TLS (TLS_TYPE=ECC)** 启动服务器将通知数据库服务器仅接受使用 ECC 加密或简单加密的连接。RSA 和 RSA FIPS 连接都失败并且无加密要求的连接会使用简单加密。

如果想要数据库服务器接受通过 TCP/IP 的加密连接，但还希望能够从本地计算机上通过共享内存连接到数据库，则可在启动数据库服务器时指定 **-es** 选项以及 **-ec** 选项。请参见“-es 服务器选项”一节第 174 页。

dbecc11.dll 和 *dbrsa11.dll* 文件均包含用于加密和解密的 ECC 和 RSA 代码。文件 *dbfips11.dll* 含有用于 FIPS 认可的 RSA 算法的代码。连接数据库服务器时，如果找不到合适的文件或者发生错误，则会在数据库服务器消息窗口中显示一条消息。如果无法启动指定类型的加密，则该服务器不启动。

客户端和服务器的加密设置必须匹配，否则连接将失败，但以下情况除外：

- 如果对数据库服务器指定了 **-ec SIMPLE**，但是没有指定 **-ec NONE**，则不要求加密的连接可以连接并自动使用简单加密。

- 如果数据库服务器指定了 RSA，客户端指定了 FIPS，或者相反，则连接将成功。这时，[Encryption] 连接属性将返回数据库服务器所指定的值。

另请参见

- “启动支持传送层安全的数据库服务器”一节第 1042 页
- “Encryption 连接参数 [ENC]”一节第 265 页
- “-ek 数据库选项”一节第 238 页
- “-ep 服务器选项”一节第 173 页
- “-es 服务器选项”一节第 174 页
- “DatabaseKey 连接参数 [DBKEY]”一节第 259 页

示例

以下示例显示允许不使用加密的连接和使用简单加密的连接。

```
dbsrv11 -ec NONE,SIMPLE -x tcpip c:\mydemo.db
```

以下示例启动一个使用椭圆曲线服务器证书 *eccserver.id* 的数据库服务器。

```
dbsrv11 -ec TLS(TLS_TYPE=ECC;IDENTITY=eccserver.id;IDENTITY_PASSWORD=test) -x tcpip c:\mydemo.db
```

以下示例启动一个使用 RSA 服务器证书 *rsaserver.id* 的数据库服务器。

```
dbsrv11 -ec TLS(TLS_TYPE=RSA;IDENTITY=rsaserver.id;IDENTITY_PASSWORD=test) -x tcpip c:\mydemo.db
```

以下示例启动一个使用 FIPS 认可的 RSA 服务器证书 *rsaserver.id* 的数据库服务器。

```
dbsrv11 -ec TLS(TLS_TYPE=RSA;FIPS=Y;IDENTITY=rsaserver.id;IDENTITY_PASSWORD=test) -x tcpip c:\mydemo.db
```

-ep 服务器选项

在启动高度加密的数据库后，提示用户提供加密密钥。

语法

```
{ dbsrv11 | dbeng11 } -ep ...
```

适用于

所有操作系统和数据库服务器。

注释

-ep 选项指示数据库服务器显示一个窗口，提示用户在其中输入加密密钥，在命令行上启动数据库需要此加密密钥。该服务器选项能避免以明文形式显示加密密钥，因而更加安全可靠。

如果与服务器一起使用，则当满足以下所有条件时，会提示用户输入加密密钥：

- 指定了 -ep 选项

- 服务器为 Windows 个人服务器，或服务器刚刚启动
- 启动数据库需要密钥
- 该服务器或者不是 Windows 服务，或者是一项开启了与桌面交互选项的 Windows 服务
- 该服务器不是守护程序 (Unix)

若要保证客户端应用程序与数据库服务器之间通信包的安全，可使用 `-ec` 服务器选项和传送层安全。请参见“[传送层安全](#)”第 1031 页。

另请参见

- “[启动支持传送层安全的数据库服务器](#)”一节第 1042 页
- “[-ec 服务器选项](#)”一节第 171 页
- “[-ek 数据库选项](#)”一节第 238 页
- “[Encryption 连接参数 \[ENC\]](#)”一节第 265 页
- “[DatabaseKey 连接参数 \[DBKEY\]](#)”一节第 259 页

示例

启动 `myencrypted.db` 数据库时，提示用户输入加密密钥：

```
dbsrv11 -ep -x tcpip myencrypted.db
```

-es 服务器选项

允许在共享内存上进行未加密的连接。

语法

```
{ dbsrv11 | dbeng11 } -ec encryption-options -es ...
```

适用于

所有服务器和操作系统，Windows Mobile 除外。

注释

仅当使用 `-ec` 选项指定时，此选项才有效。`-es` 选项指示数据库服务器通过共享内存允许未加密的连接。通过 TCP/IP 的连接必须使用 `-ec` 选项指定的加密类型。此选项对于想要远程客户端使用加密连接的情况非常有用，但出于性能方面的原因，最好从本地计算机使用未加密的连接访问数据库。

另请参见

- “[-ec 服务器选项](#)”一节第 171 页
- “[启动支持传送层安全的数据库服务器](#)”一节第 1042 页

示例

以下示例指定允许使用简单加密的连接以及未加密的共享内存连接。

```
dbsrv11 -ec SIMPLE -es -x tcpip c:\mydemo.db
```


-f 恢复选项

强制在事务日志丢失后启动数据库服务器。

语法

```
{ dbsrv11 | dbeng11 } -f ...
```

适用于

所有操作系统和数据库服务器。

注释

小心
此选项只在恢复情况下使用。

如果没有事务日志，数据库服务器会执行数据库的检查点恢复，然后关闭—它不再继续运行。然后您可以在不使用 -f 选项的情况下重新启动该数据库服务器，以进行正常操作。

如果数据库所在的目录中有事务日志，则数据库服务器会执行检查点恢复，并使用该事务日志进行恢复，然后关闭—服务器不再继续运行。然后您可以在不使用 -f 选项的情况下重新启动该数据库服务器，以进行正常操作。

启动服务器时指定高速缓存大小可缩短恢复时间。

另请参见

- “在特殊模式下运行” 一节第 46 页
- “备份和数据恢复” 第 817 页

示例

以下命令强制数据库服务器启动并对数据库 *mydatabase.db* 执行恢复：

```
dbeng11 mydatabase.db -f
```

-fc 服务器选项

指定包含文件系统已满回调函数的 DLL（在 Unix 上为共享对象）的文件名。

语法

```
{ dbsrv11 | dbeng11 } -fc filename ...
```

适用于

所有操作系统和数据库服务器。

注释

此选项可用于在遇到文件系统已满的情况时通知用户并可能采取纠正操作。如果使用 -fc 选项，则数据库服务器在启动过程中会尝试装载指定的 DLL，并解析回调函数的入口点。如果 SQL

Anywhere 数据库服务器找不到 DLL 和入口点，则数据库服务器返回一个错误并关闭。DLL 由用户提供，可以使用回调来实现包括调用您提供的用来诊断或纠正操作的批处理文件（在 Unix 上为 shell 脚本）等功能。另外，回调函数本身也可以执行这样的操作。

磁盘已满回调函数的示例位于 *samples-dir\SQLAnywhere\DiskFull* 中。

有关 *samples-dir* 的信息，请参见“[示例目录](#)”一节第 368 页。

SQL Anywhere 在搜索其它 DLL 和文件的位置搜索回调函数 DLL。

有关 SQL Anywhere 在何处搜索文件的详细信息，请参见“[SQL Anywhere 如何定位文件](#)”一节第 370 页。

当数据库服务器检测到磁盘已满情况时，它会调用回调函数（如果已提供了一个此类函数）并向它传递以下信息：

- 在其中触发了此条件的 `dbspace` 的名称
- 来自失败操作的操作系统特定的错误代码

调用 `xp_out_of_disk` 所返回的代码指示应中止还是重试导致出现此情况的操作。如果返回一个非零值，则中止操作，否则重试操作。只要它返回零并且文件系统操作失败，就会反复调用回调函数。

在 Microsoft Windows 平台上，如果数据库服务器已经启动并显示数据库服务器消息窗口（既未指定 `-qi`，也未指定 `-qw`），并且未提供回调 DLL，则当出现磁盘占满的情况时，会显示一个窗口。此窗口包含 `dbspace` 名称和错误代码，并允许用户选择是重试还是中止导致磁盘已满的操作。

在所有其它操作系统上，如果未指定 `-fc` 并且遇到磁盘已满的情况，则会发生致命错误。

您可以创建系统事件，监视用来保留数据库文件、日志文件或临时文件的设备的可用磁盘空间，并在磁盘空间不足时提醒管理员。

请参见“[CREATE EVENT 语句](#)”一节《[SQL Anywhere 服务器 - SQL 参考](#)》。

另请参见

- “[使用回调函数](#)”一节《[SQL Anywhere 服务器 - 编程](#)》
- “[了解系统事件](#)”一节第 868 页
- “[max_temp_space 选项 \[数据库\]](#)”一节第 521 页
- “[temp_space_limit_check 选项 \[数据库\]](#)”一节第 552 页

示例

当数据库服务器启动时，它会尝试装载 `diskfull.dll` DLL。

```
dbeng11 -fc diskfull.dll
```

-fips 服务器选项

要求仅使用 FIPS 认可的算法来对数据库和通信进行高度加密。

语法

```
{ dbsrv11 | dbeng11 } -fips ...
```

适用于

Windows

注释

指定此选项会强制所有服务器加密均使用 FIPS 认可的算法。此选项适用于高度数据库加密、客户端 / 服务器传送层安全性和 Web 服务传送层安全性。当指定了 `-fips` 选项时，您仍然可以使用未加密的连接和数据库，但不能使用简单加密。

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》。

对于高度数据库加密，`-fips` 选项导致新的数据库使用 `AES_FIPS` 类型，即使在 `CREATE DATABASE` 语句的 `ALGORITHM` 子句中指定了 `AES` 也是如此。

在以 `-fips` 启动数据库服务器时，您可以运行用 `AES`、`AES256`、`AES_FIPS` 或 `AES256_FIPS` 高度加密方法加密的数据库，但不能运行用简单加密方法加密的数据库。指定 `-fips` 时，也可以在服务器上启动未加密的数据库。

必须在用于运行使用 `AES_FIPS` 或 `AES256_FIPS` 加密的数据库的所有计算机上都安装 SQL Anywhere 安全性组件。

对于 SQL Anywhere 传送层安全，`-fips` 选项会使服务器使用 FIPS 认可的 RSA 加密编码器（即使指定了 `RSA` 也是如此）。如果指定了 `ECC`，则会因 FIPS 认可的椭圆曲线算法不可用而发生错误。

对于 Web 服务的传送层安全，`-fips` 选项将使服务器使用 `HTTPS FIPS`（即使指定了 `HTTPS` 也是如此）。

指定 `-fips` 时，`ENCRYPT` 和 `HASH` 函数将使用 FIPS 认可的 RSA 加密编码器，并且口令散列将使用 `SHA-256 FIPS` 算法而非 `SHA-256` 算法。

另请参见

- “高度加密”一节第 1019 页
- “传送层安全”第 1031 页
- “加密 SQL Anywhere Web 服务”一节第 1046 页
- “`-ec` 服务器选项”一节第 171 页
- “`ENCRYPT` 函数 [String]”一节《SQL Anywhere 服务器 - SQL 参考》
- “`HASH` 函数 [String]”一节《SQL Anywhere 服务器 - SQL 参考》

-ga 服务器选项

在最后的非 HTTP 客户端连接断开后卸载数据库。

语法

```
{ dbsrv11 | dbeng11 } -ga ...
```

适用于

所有操作系统。

注释

在网络服务器上指定此选项会导致每个数据库在最后一个非 HTTP 客户端连接断开后被卸载。不仅会在最后一个非 HTTP 连接断开后卸载各个数据库，还会在停止最后一个数据库时关闭数据库服务器。

如果只有与数据库的连接是 HTTP 连接而且该数据库配置为自动停止，则当该 HTTP 连接断开时，不会卸载数据库。同样，如果指定 **-ga** 选项并且数据库具有 HTTP 连接和命令序列或 TDS 连接，则当最后一个命令序列或 TDS 连接断开时，该数据库就会自动停止，并且所有 HTTP 连接都将断开。

另请参见

- “重建数据库”一节 《SQL Anywhere 服务器 - SQL 的用法》
- “AutoStop 连接参数 [ASTOP]”一节第 251 页

-gb 服务器选项

设置服务器进程优先级。

Windows 语法

```
{ dbsrv11 | dbeng11 } -gb { idle | normal | high | maximum } ...
```

Unix 语法

```
{ dbsrv11 | dbeng11 } -gb level ...
```

适用于

Windows、Unix、Mac OS X

注释

此选项设置服务器进程优先级。

在 Windows 上，**normal** 和 **high** 为常用设置。提供值 **idle** 是为了使值集完整，而 **maximum** 可能会影响计算机的运行。

在 Unix 上，**level** 是一个介于 -20 和 19 之间的整数。Unix 上的缺省值与父进程的精确值相同。较低的 **level** 值代表更加合适的调度优先级。有关设置精确值的所有限制均适用于 **-gb** 选项。例如，在大多数 Unix 平台上，只有 **root** 用户能够降低进程的优先级（例如，将其从 0 更改为 -1）。

-gc 服务器选项

设置检查点之间的最大间隔。

语法

```
{ dbsrv11 | dbeng11 } -gc minutes ...
```

适用于

所有操作系统和数据库服务器。

注释

设置数据库服务器在不对各个数据库执行检查点操作的情况下运行的最长时间（以分钟为单位）。

缺省值为 `checkpoint_time` 数据库选项的设置，该值缺省为 60 分钟。如果输入值 0，则使用缺省值 60 分钟。

检查点通常比指定时间发生得更频繁。

请参见“数据库服务器如何确定何时进行检查点操作”一节第 853 页。

另请参见

- “`checkpoint_time` 选项 [数据库]”一节第 488 页
- “了解检查点日志”一节第 17 页
- “数据库服务器如何确定何时进行检查点操作”一节第 853 页

-gd 服务器选项

设置启动或停止数据库所需的权限。

语法

```
{ dbsrv11 | dbeng11 } -gd { DBA | all | none } ...
```

适用于

所有操作系统和数据库服务器。

注释

用户需要具有此权限才能使服务器装载新数据库文件，或停止正在运行的数据库服务器上的数据库。可以是下列级别之一：

- **DBA** 只有拥有 DBA 权限的用户才能启动或停止数据库。
- **all** 所有用户都可以启动或停止数据库。
- **none** 除数据库服务器自行启动和停止外，不允许启动和停止数据库。

个人数据库服务器的缺省设置为 `all`，而网络数据库服务器的缺省设置为 `DBA`。大小写语法均可接受。

注意，此选项设置为 `DBA` 时，客户端应用程序必须已经与服务器连接才能启动或停止数据库。仅为新连接提供 `DBA` 用户 ID 和口令是不够的。

使用 `StartDBPermission` 服务器属性，可获得 `-gd` 选项的设置：

```
SELECT PROPERTY ( 'StartDBPermission' );
```

另请参见

- [“权限概述”一节第 428 页](#)

示例

下面这组步骤说明了怎样将 `-gd` 选项用于网络数据库服务器。

1. 启动网络数据库服务器:

```
dbsrv11 -x tcpip -su mypwd -n myserver -gd DBA
```

2. 从 Interactive SQL 连接到实用程序数据库:

```
dbisql -c "UID=DBA;PWD=mypwd;ENG=myserver;DBN=utility_db"
```

3. 启动数据库:

```
START DATABASE demo  
ON myserver;
```

4. 连接到已启动的数据库:

```
CONNECT  
TO myserver  
DATABASE demo  
USER DBA IDENTIFIED BY sql;
```

-ge 服务器选项

设置外部函数的堆栈大小。

语法

```
{ dbsrv11 | dbeng11 } -ge integer ...
```

适用于

Windows

注释

为运行外部函数的线程设置堆栈大小（以字节为单位）。缺省值为 32 KB。

另请参见

- [“控制线程行为”一节第 48 页](#)

-gf 服务器选项

服务器禁止触发触发器。

语法

```
{ dbsrv11 | dbeng11 } -gf ...
```

适用于

所有操作系统和数据库服务器。

注释

-gf 服务器选项指示服务器禁止触发器的触发。

另请参见

- “[fire_triggers 选项 \[兼容性\]](#)” 一节第 504 页
- “[触发器简介](#)” 一节 《SQL Anywhere 服务器 - SQL 的用法》

-gk 服务器选项

设置使用 dbstop 停止网络服务器和个人服务器所需的权限。

语法

```
{ dbsrv11 | dbeng11 } -gk { DBA | all | none } ...
```

适用于

所有操作系统和数据库服务器。

注释

可用值如下：

- **DBA** 只有拥有 DBA 权限的用户才可使用 dbstop 来停止服务器。这是网络服务器的缺省设置。
- **all** 所有用户都可使用 dbstop 来停止服务器。这是个人服务器的缺省设置。
- **none** 不能使用 dbstop 来停止该服务器。

大小写语法均可接受。

另请参见

- “[停止服务器实用程序 \(dbstop\)](#)” 一节第 783 页

-gl 服务器选项

设置使用 LOAD TABLE 装载数据和使用 UNLOAD 或 UNLOAD TABLE 卸载数据所需的权限。

语法

```
{ dbsrv11 | dbeng11 } -gl { DBA | all | none } ...
```

适用于

所有操作系统和数据库服务器。

注释

使用 UNLOAD TABLE 或 UNLOAD 语句将数据放置到数据库服务器计算机上的文件中，使用 LOAD TABLE 语句从数据库服务器计算机中读取文件。

要控制使用这些语句对文件系统进行访问，可使用 `-gl` 服务器选项来控制使用这些语句所需的数据库权限级别。

可用值如下：

- **DBA** 只允许拥有 DBA 权限的用户从数据库装载或卸载数据。
- **all** 所有用户都可从该数据库装载或卸载数据。
- **none** 不能卸载或装载数据。

大小写语法均可接受。

对于非 Unix 操作系统上的个人数据库服务器，缺省设置为 **all**；对于网络数据库服务器和 Unix 个人服务器，缺省设置为 **DBA**。这些设置反映了这样一个事实，即在非 Unix 平台上，个人数据库服务器运行在当前计算机上，因此用户已经具有访问该文件系统的权限。

另请参见

- [“LOAD TABLE 语句”一节 《SQL Anywhere 服务器 - SQL 参考》](#)
- [“UNLOAD 语句”一节 《SQL Anywhere 服务器 - SQL 参考》](#)

-gm 服务器选项

限制服务器的并发连接数。

语法

```
{ dbsrv11 | dbeng11 } -gm integer ...
```

适用于

所有操作系统和数据库服务器。

注释

定义服务器的连接限制。如果此数字大于授权和内存约束所允许的数量，则该值无效。

数据库服务器允许在连接限制之上建立一个额外的 DBA 连接，以便允许用户在出现紧急情况时使用 DBA 授权连接到服务器并删除其它连接。

-gn 服务器选项

设置数据库服务器可同时执行的最大任务数。

语法

```
{ dbsrv11 | dbeng11 } -gn integer ...
```

适用于

所有操作系统和数据库服务器。

注释

该选项设置数据库服务器的最大进程并发水平。它将限制数据库服务器可并发执行的任务数（包括用户请求和系统请求）。如果数据库服务器在达到此限制时接收到其它请求，则新请求必须等待，直至正在执行的任务完成。

未调度请求数和活动请求数之和的最大值受 `-gm` 服务器选项的限制，该选项限制与服务器的连接数。

将 `-gn` 值设置过高会导致出现错误，因为数据库服务器的系统资源由该较大 `-gn` 值消耗。

对于网络数据库服务器和个人数据库服务器，除 Windows Mobile 外（其缺省值为 3），缺省值均为 20 个活动任务，可同时执行的活动任务数取决于正在使用的数据库服务器线程数和逻辑处理器数。

数据库服务器的内核将任务用作调度单位。执行任何用户请求至少需要一个任务。不过，请求可能会为自身得以执行而调度更多任务。上述情况的一个示例是：如果请求涉及执行某个外部过程或函数（Java、Perl、CLR 等），其执行结果会使多个数据库请求依次返回到数据库服务器。

当涉及查询内并行机制时，每个并行执行的访问计划组件均为一个任务。这些任务将被视为单独的请求进行计数，直至达到 `-gn` 限制。不过，为查询内并行机制而创建的任务不会在用于跟踪活动和非活动请求数的数据库属性中得以反映。

小心

将为每个数据库服务器任务分配由 `-gss` 指定大小的堆栈，且任务的最大数目由 `-gn` 选项指定。如果将 `-gss` 和 `-gn` 同时设置为一个高值，则数据库服务器可能无法启动，或高速缓存的大小可能受到明显限制。例如，如果在启动数据库服务器时指定 `-gss` 为 16M，`-gn` 为 100，则仅为堆栈保留 1.6 GB 的内存。

另请参见

- “SQL Anywhere 中的线程” 一节第 47 页
- “设置数据库服务器的进程并发水平” 一节第 50 页
- “`max_query_tasks` 选项 [数据库]” 一节第 519 页
- “`-gm` 服务器选项” 一节第 182 页
- “`-gm` 服务器选项” 一节第 182 页
- “`-gtc` 服务器选项” 一节第 186 页

-gp 服务器选项

设置允许的最大数据库页面大小。

语法

```
{ dbsrv11 | dbeng11 } -gp { 2048 | 4096 | 8192 | 16384 | 32768 } ...
```

适用于

所有操作系统和数据库服务器。

注释

不能装载页面大小大于服务器页面大小的数据库文件。此选项显式设置服务器的页面大小（以字节为单位）。

如果不使用此选项，则使用命令行中第一个数据库的页面大小。

在所有平台上，如果不使用此选项，并且在不装载数据库的情况下启动服务器，则缺省值为 4096。

另请参见

- [“表大小和页面大小”一节《SQL Anywhere 服务器 - SQL 的用法》](#)
- [“设置最大页面大小”一节第 46 页](#)

-gr 服务器选项

设置从系统故障中恢复所用的最长时间（以分钟为单位）。

语法

```
{ dbsrv11 | dbeng11 } -gr minutes ...
```

适用于

所有操作系统和数据库服务器。

注释

当数据库服务器运行多个数据库时，使用由第一个启动的数据库指定的恢复时间，除非被此选项替换。

-gr 选项指定的值会指示数据库服务器多长时间执行一次检查点操作。例如，如果将 -gr 设置为 5，则数据库服务器会频繁尝试执行检查点操作，使恢复时间不会长于 5 分钟。不过，如果必须进行恢复，则即使恢复时间大于 -gr 所指定的时间长度，系统也要完成恢复过程。缺省值为 `recovery_time` 数据库选项的设置，该值缺省为 2 分钟。

恢复时间既包括数据库的估计恢复时间，也包括数据库的估计检查点操作时间。

另请参见

- [“`recovery_time` 选项 \[数据库\]”一节第 538 页](#)
- [“数据库服务器如何确定何时进行检查点操作”一节第 853 页](#)

-gss 服务器选项

设置服务器中每个内部执行线程的堆栈大小。

语法

```
{ dbsrv11 | dbeng11 } -gss { integer[ k | m ] } ...
```

适用于

所有操作系统和服务器。对于 Windows，在 Windows XP 和更高版本上支持此选项。

注释

内部执行的线程数由 `-gn` 选项控制，缺省值为 20。`-gss` 选项允许在内存有限的环境中降低数据库服务器内存的使用。

`size` 是要使用的内存量（以字节为单位）。分别使用 `k` 或 `m` 将单位指定为千字节或兆字节。

小心

将为每个数据库服务器任务分配由 `-gss` 指定大小的堆栈，且任务的最大数目由 `-gn` 选项指定。如果将 `-gss` 和 `-gn` 同时设置为一个高值，则数据库服务器可能无法启动，或高速缓存的大小可能受到明显限制。例如，如果在启动数据库服务器时指定 `-gss` 为 16M，`-gn` 为 100，则仅为堆栈保留 1.6 GB 的内存。

在 Windows XP 和更高版本上，数据库服务器使用的缺省堆栈大小在 32 位操作系统上为 1 MB，在 64 位操作系统上为 4 MB。数据库服务器使用的最大堆栈大小在 32 位操作系统上为 16 MB，在 64 位操作系统上为 256 MB。在 Windows 2000 上忽略此选项。

在 Unix 上，每个内部执行线程的最小堆栈大小的缺省值为 500 KB，最大堆栈大小为 4 MB。

Pocket PC 2003 和更高版本支持此选项。在所支持的 Windows Mobile 平台上，最小堆栈大小的缺省值为 64 KB，最大堆栈大小为 512 KB。在早期版本的 Windows Mobile 平台上，每线程保留 1MB 的地址空间。

另请参见

- “SQL Anywhere 中的线程”一节第 47 页

-gt 服务器选项

设置最多可使用的物理处理器数（不超过授权使用的最大值）。此选项仅在多处理器系统中有用。

语法

```
{ dbsrv11 | dbeng11 } -gt integer ...
```

适用于

Windows（除 Windows Mobile）、Linux 和 Solaris。

注释

个人数据库服务器能够使用的处理器数量始终限制为一个。对于每存取设备授权，网络数据库服务器使用计算机上可用的所有 CPU（缺省设置）。对于基于 CPU 的授权，网络数据库服务器仅使用您得到授权的处理器。网络数据库服务器可以使用的 CPU 数量还可能会受到 SQL Anywhere 版本的影响。请参见“版本和授权”一节《SQL Anywhere 11 - 简介》。

若指定 `-gt` 选项的值，则数据库服务器将调整其处理器掩码（如果在该硬件平台受支持），以将数据库服务器限制为仅使用设置数量的物理处理器运行。如果许可数据库服务器使用 n 个处理器，则缺省情况下，服务器将使用 n 个物理处理器的所有逻辑处理器（超线程和内核）运行。可通过 `-gtc` 选项对此进行进一步限制。

`-gt` 选项的有效值介于 1 和以下各项中的最小值之间：

- 计算机上物理处理器的数量
- 授权服务器使用的最大 CPU 数（如果 CPU 授权有效）

如果指定的 `-gt` 值在此范围之外，则会强制使用下限或上限值。对于个人数据库服务器 (`dbeng11`)，服务器使用的 `-gt` 值为 1。

另请参见

- “`-gn` 服务器选项”一节第 182 页
- “`-gtc` 服务器选项”一节第 186 页
- “SQL Anywhere 中的线程”一节第 47 页

-gtc 服务器选项

控制数据库服务器允许的处理器最大并发数。

语法

```
{ dbsrv11 | dbeng11 } -gtc logical-processors-to-use ...
```

适用于

在 Intel 兼容的 x86 和 x64 平台上执行的 Linux、Solaris 和 Windows 操作系统（Windows Mobile 除外）。

注释

启动数据库服务器时，数据库服务器检测到的物理处理器数和逻辑处理器数将显示在数据库服务器消息窗口中。

物理处理器有时被称为 **packages**（包）或 **dies**（芯片），它们是计算机的 CPU。如果物理处理器支持超线程或其自身配置为**多处理器**（通常称为多核处理器），则存在其它逻辑处理器。操作系统在逻辑处理器上调度线程。

`-gtc` 选项允许您指定可由数据库服务器使用的逻辑处理器数。其作用是限制服务器启动时创建的数据库服务器线程数。这将限制任意时刻可并发执行的活动数据库服务器任务数。缺省情况下，创建的线程数为所有授权物理处理器上的逻辑处理器数加上 1。

缺省情况下，数据库服务器允许以并发方式使用每个授权使用的物理处理器上的所有逻辑处理器（内核或超线程）。例如，在支持超线程的单 CPU 系统上，缺省情况下，数据库服务器允许在一个物理处理器上并发运行两个线程。如果指定 `-gtc` 选项，并且要使用的逻辑处理器数少于授权物理处理器可用逻辑处理器数的总和，则数据库服务器将基于轮转调度分配方式分配逻辑处理器。将 `-gtc` 选项指定为 1 会隐式禁用查询内并行机制（并行处理各个查询）。也可以使用 `max_query_tasks` 选项显式限制或彻底禁用查询内并行机制。请参见“[max_query_tasks 选项 \[数据库\]](#)”一节第 519 页。

另请参见

- “[-gn 服务器选项](#)”一节第 182 页
- “[-gt 服务器选项](#)”一节第 185 页
- “[查询执行期间的并行](#)”一节《[SQL Anywhere 服务器 - SQL 的用法](#)》
- “[SQL Anywhere 中的线程](#)”一节第 47 页

示例

对于基于 Windows 的 SMP 计算机，请考虑以下示例。在每种情况下，都假设有一个 4 处理器的系统，每个物理处理器上都包含双核（因此总共有 8 个逻辑处理器）。物理处理器用字母进行标识，逻辑处理器（本例中为核）用数字进行标识。此 4 处理器系统因此便具有处理单元 A0、A1、B0、B1、C0、C1、D0 和 D1。

方案	网络数据库服务器设置
单 CPU 授权或指定 <code>-gt 1</code>	<ul style="list-style-type: none"> ● <code>-gt 1</code> ● <code>-gtc 2</code> ● <code>-gn 20</code> <p>这些线程可在 A0 和 A1 上执行。</p>
无 CPU 授权限制，指定了 <code>-gtc 5</code>	<ul style="list-style-type: none"> ● <code>-gt 4</code> ● <code>-gtc 5</code> ● <code>-gn 20</code> <p>这些线程可在 A0、A1、B0、C0 和 D0 上执行。</p>
具有 3 CPU 授权并指定了 <code>-gtc 5</code> 的数据库服务器	<ul style="list-style-type: none"> ● <code>-gt 3</code> ● <code>-gtc 5</code> ● <code>-gn 20</code> <p>这些线程可在 A0、A1、B0、B1 和 C0 上执行。</p>
无 CPU 授权限制，指定了 <code>-gtc 1</code>	<ul style="list-style-type: none"> ● <code>-gt 4</code> ● <code>-gtc 1</code> ● <code>-gn 20</code> <p>这些线程只能在 A0 上执行。</p>

-gu 服务器选项

设置实用程序命令的权限级别。

语法

```
{ dbsrv11 | dbeng11 } -gu { all | none | DBA | utility_db } ...
```

适用于

所有操作系统和数据库服务器。

注释

针对 CREATE DATABASE 和 DROP DATABASE 等实用程序命令设置权限级别。可设置为下列级别之一：utility_db、all、none 或 DBA。缺省值为 DBA。

utility_db 级别将这些命令的使用者仅限于能连接到实用程序数据库的用户。all、none 和 DBA 级别分别允许所有用户、无用户或拥有 DBA 权限的用户执行实用程序命令。

另请参见

- “指定执行文件管理语句所需的权限”一节第 30 页

-im 服务器选项

在内存中运行数据库服务器，减少或消除到磁盘的写入。

语法

```
{ dbsrv11 | dbeng11 } -im { c | nw } ...
```

适用于

所有操作系统和数据库服务器。

需要单独授予许可的组成部分

内存模式需要一个单独的许可。请参见“单独授权的组件”一节《SQL Anywhere 11 - 简介》。

注释

此功能在可用内存量较大的系统上非常有用，通常高速缓存可以容纳所有的数据库文件。可供使用的内存模式有两个：

- **仅检查点 (-im c)** 当以仅检查点模式运行时，数据库服务器不使用事务日志，所以您不能恢复到最近提交的事务。但是，由于启用了检查点日志，所以数据库可以恢复到最近的检查点。通常，当您运行不包含事务日志的数据库时，数据库服务器仍在提交时执行检查点操作，这会影晌性能。但是当以仅检查点模式运行数据库服务器时，数据库服务器在每次提交后不执行检查点操作。

对于希望增加性能的应用程序，此模式非常有用，同时允许丢失在最近检查点之后所提交的事务。

当以仅检查点模式运行时，以下限制适用：

1. 无事务日志。
2. 无临时文件。
3. 允许根据需要以及按照数据库服务器的正常检查点频率来设置检查点。
4. 仅在检查点处将脏页刷新到磁盘中。

- **永不写入 (-im nw)** 当以永不写入模式运行时，已提交事务将不会写入磁盘上的数据库文件中。如果数据库关闭或崩溃，所有更改将丢失，因此，数据库文件将始终停留在其原始状态。允许请求扩展或创建新 **dbspace**，但更改不会反映在数据库文件中。可以创建并使用新的 **dbspace**，但是它们不会被写入磁盘。在永不写入模式下执行备份毫无用处，因为对系统 **dbspace** 所做的任何更改都不会写入到文件。

当以永不写入模式运行时，以下限制适用：

1. 无事务日志。
2. 无检查点日志。
3. 无临时文件。
4. 永不将脏数据库页刷新到磁盘中。
5. 永不修改原始数据库文件。

因为从不将更改写入到原始数据库文件，所以，如果需要当前数据库内容的持久副本，则必须使用 **dbunload** 实用程序或 **UNLOAD TABLE** 语句。还可以使用 **SQL** 查询来检索更改，但随后必须手动将这些更改写入到数据库文件。

从内存模式获得的性能优势取决于应用程序负载和 I/O 子系统的速度。对于插入或更新大量数据的应用程序以及频繁执行提交和检查点操作的应用程序，可获得最大性能优势。

通常，内存模式的性能等同于或优于使用事务性全局临时表时的性能。对于主要用于查询数据库的应用程序，所获得的性能优势最小。通常，在使用内存模式时，可通过将高速缓存预增加到足以容纳数据库文件全部预期内容来实现其最佳性能。这消除了运行应用程序时增量增加高速缓存所带来的大部分开销。

小心

由于在永不写入模式下不刷新高速缓存中的页面，因此当数据库中的数据量增长过大时可能会耗尽可用高速缓存。当发生这种情况时，**SQL Anywhere** 会发出错误消息并停止处理请求。因此，应小心使用永不写入模式，并始终保持足够大的高速缓存以容纳应用程序可能使用的预期的全部有效页集。由于在“仅检查点”模式中持续出现检查点，所以与“永不写入”模式相比，它降低了服务器用尽可用高速缓存的风险。

对于 **LOAD TABLE** 语句和某些 **ALTER TABLE** 语句，检查点日志用于部分撤消语句失败的作用或是从错误恢复。在永不写入模式下，将不创建检查点日志，如果某些语句失败或是发生错误，则无法部分撤消这些语句的作用。表中会保留不正确或不完整的数据。请参见“[了解检查点日志](#)”一节第 17 页。

另请参见

- “单独授权的组件”一节 《SQL Anywhere 11 - 简介》
- “-c 服务器选项”一节第 158 页
- “使用内存模式”一节 《SQL Anywhere 服务器 - SQL 的用法》

-k 服务器选项

控制性能监控器统计信息的收集。

语法

```
{ dbsrv11 | dbeng11 } -k ...
```

适用于

所有操作系统和数据库服务器。

注释

缺省情况下，数据库服务器将收集 [性能监控器] 统计信息。

如果在启动数据库服务器时指定 -k，则服务器不收集 [性能监控器] 统计信息。-k 选项不会影响查询优化程序所用的列统计信息收集。

仅当数据库服务器运行于多处理器计算机上，并且测试显示可以改进性能时，才应使用该选项。对于大多数负载而言，其优点是微不足道的，因此建议不使用该选项。当禁用性能计数器时，该信息无法用于分析性能问题。

也可使用 sa_server_option 系统过程更改 [性能监控器] 统计信息收集的设置。请参见 “sa_server_option 系统过程”一节 《SQL Anywhere 服务器 - SQL 参考》。

另请参见

- “-ks 服务器选项”一节第 193 页
- “-ksc 服务器选项”一节第 193 页
- “-ksd 服务器选项”一节第 193 页
- “使用 Sybase Central 性能监控器监控统计信息”一节 《SQL Anywhere 服务器 - SQL 的用法》

-kl 服务器选项

指定 Kerberos GSS-API 库（在 Unix 上为共享对象）的文件名称，并允许与数据库服务器建立采用 Kerberos 验证的连接。

语法

```
{ dbsrv11 | dbeng11 } -kl GSS-API-library-file ...
```

适用于

所有操作系统，Windows Mobile 除外。

注释

该选项用于指定 Kerberos GSS-API 的位置和名称。仅当 Kerberos 客户端对 Kerberos GSS-API 库使用的文件名与缺省文件名不同时，或运行数据库服务器的计算机上安装了多个 GSS-API 库时，该选项才是必需项。必须已经安装和配置 Kerberos 客户端，并且数据库服务器不能使用 SSPI。

指定该选项允许对数据库服务器进行 Kerberos 验证。

另请参见

- “-kr 服务器选项” 一节第 191 页
- “-krb 服务器选项” 一节第 192 页
- “Kerberos 连接参数 [KRB]” 一节第 270 页
- “Kerberos 验证” 一节第 107 页
- “GRANT 语句” 一节 《SQL Anywhere 服务器 - SQL 参考》

示例

以下命令将启动使用 `libgssapi_krb5.so` 共享对象进行 Kerberos 验证的数据库服务器。

```
dbsrv11 -kl libgssapi_krb5.so -n my_server_princ /opt/myapp/kerberos.db
```

-kr 服务器选项

指定 Kerberos 服务器主体的域并允许与数据库服务器建立采用 Kerberos 验证的连接。

语法

```
{ dbsrv11 | dbeng11 } -kr server-realm ...
```

适用于

所有操作系统，Windows Mobile 除外。

注释

该选项用于指定 Kerberos 服务器主体的域。通常，数据库服务器进行 Kerberos 验证所用的主体为 `server-name@default-realm`，其中 `default-realm` 是为 Kerberos 客户端配置的缺省域。如果希望服务器主体使用其它域而非缺省域，可使用该选项，这时，所用的服务器主体是 `server-name@server-realm`。

指定该选项允许对数据库服务器进行 Kerberos 验证。

另请参见

- “-kl 服务器选项” 一节第 190 页
- “-krb 服务器选项” 一节第 192 页
- “Kerberos 连接参数 [KRB]” 一节第 270 页
- “Kerberos 验证” 一节第 107 页
- “GRANT 语句” 一节 《SQL Anywhere 服务器 - SQL 参考》

示例

以下命令将启动接受 Kerberos 登录并使用主体 `my_server_princ@MYREALM` 进行验证的数据库服务器。

```
dbeng11 -kr MYREALM -n my_server_princ C:\kerberos.db
```

-krb 服务器选项

允许与数据库服务器建立采用 Kerberos 验证的连接。

语法

```
{ dbsrv11 | dbeng11 } -krb ...
```

适用于

所有操作系统，Windows Mobile 除外。

注释

该选项允许对数据库服务器进行 Kerberos 验证。必须为数据库服务器指定一个或多个 `-krb`、`-kl` 和 `-kr` 选项以便能够使用 Kerberos 验证客户端。

必须事先在客户端以及数据库服务器计算机上安装和配置 Kerberos 客户端，之后才能使用 Kerberos 验证。此外，主体 `server-name@REALM` 必须已经存在于 Kerberos KDC 中，且主体 `server-name@REALM` 的 keytab 必须已经安全地抽取到数据库服务器计算机上的 keytab 文件中。如果指定了 `-krb` 选项，但尚未执行此设置，则数据库服务器不会启动。

注意

数据库服务器名称不能含有以下任何字符：/、\ 或 @，并且 Kerberos 不能使用由多字节字符组成的数据库服务器名称。

必须将 `login_mode` 数据库选项设置为允许 Kerberos 登录，并且必须使用 `GRANT KERBEROS LOGIN` 语句将 Kerberos 客户端主体映射到数据库用户 ID。

另请参见

- [“-kl 服务器选项”一节第 190 页](#)
- [“-kr 服务器选项”一节第 191 页](#)
- [“Kerberos 连接参数 \[KRB\]”一节第 270 页](#)
- [“Kerberos 验证”一节第 107 页](#)
- [“GRANT 语句”一节 《SQL Anywhere 服务器 - SQL 参考》](#)

示例

以下命令将为名为 `my_server_princ@MYREALM` 的数据库服务器的 Kerberos 主体，启动名为 `my_server_princ` 的数据库服务器。

```
dbsrv11 -krb -n my_server_princ C:\kerberos.db
```

-ks 服务器选项

禁止创建性能监控器用于从数据库服务器收集计数器值的共享内存。

语法

```
{ dbsrv11 | dbeng11 } -ks 0 ...
```

适用于

Windows

注释

当您指定此选项时，性能监控器不显示任何服务器、数据库或当前数据库服务器的连接统计信息。

另请参见

- “使用 Sybase Central 性能监控器监控统计信息”一节 《SQL Anywhere 服务器 - SQL 的用法》
- “-k 服务器选项”一节第 190 页
- “-ksc 服务器选项”一节第 193 页
- “-ksd 服务器选项”一节第 193 页

-ksc 服务器选项

指定性能监控器可以监控的最大连接数。

语法

```
{ dbsrv11 | dbeng11 } -ksc integer ...
```

适用于

Windows

注释

缺省情况下，性能监控器监控 10 个连接。

另请参见

- “使用 Sybase Central 性能监控器监控统计信息”一节 《SQL Anywhere 服务器 - SQL 的用法》
- “-k 服务器选项”一节第 190 页
- “-ks 服务器选项”一节第 193 页
- “-ksd 服务器选项”一节第 193 页

-ksd 服务器选项

指定性能监控器可以监控的数据库的最大数目。

语法

{ **dbsrv11** | **dbeng11** } **-ksd** *integer* ...

适用于

Windows

注释

缺省情况下，性能监控器监控两个数据库。

另请参见

- “使用 Sybase Central 性能监控器监控统计信息”一节 《SQL Anywhere 服务器 - SQL 的用法》
- “-k 服务器选项”一节第 190 页
- “-ks 服务器选项”一节第 193 页
- “-ksc 服务器选项”一节第 193 页

-m 服务器选项

在检查点完成之后截断事务日志。

语法

{ **dbsrv11** | **dbeng11** } **-m** ...

适用于

所有操作系统和数据库服务器。

注释

此选项在检查点完成之后截断事务日志，可以在关机时，也可以是按照服务器所安排的检查点的结果。

小心

当选择该选项时，对保存数据库文件的设备上的介质故障将没有保护。

此选项提供了自动限制事务日志增加的方法。检查点频率仍由 `checkpoint_time` 和 `recovery_time` 选项控制（也可以在命令行设置）。

如果要在处理需要快速响应的大容量事务时限制事务日志的大小，且事务日志的内容与恢复或复制无关，则 `-m` 选项非常有用。`-m` 选项为在没有事务日志的情况下执行操作提供了另一种方法，在这种情况下，每个 `COMMIT` 后面都需要一个检查点，因此性能会受影响。如果选择 `-m` 选项，对包含数据库文件的设备上的介质故障将没有保护。在使用 `-m` 选项之前，应考虑其它用于管理事务日志的替代方法（例如，使用 `BACKUP` 语句和事件）。

为避免出现数据库文件碎片，建议您在使用该选项时，将事务日志存放在与数据库本身不同的单独设备或分区上。

如果使用该选项，则在进行检查点操作时无法执行任何其它操作。

小心

不要对正在复制或同步的数据库使用 `-m` 选项。SQL Remote 和 MobiLink 所使用的复制和同步对事务日志信息有内在的依赖性。

另请参见

- “`-m` 数据库选项” 一节第 238 页
- “事务日志” 一节第 14 页
- “了解检查点日志” 一节第 17 页
- “事务日志实用程序 (dblog)” 一节第 792 页
- “`checkpoint_time` 选项 [数据库]” 一节第 488 页
- “`recovery_time` 选项 [数据库]” 一节第 538 页

-n 服务器选项

设置数据库服务器的名称。

语法

```
{ dbsrv11 | dbeng11 } -n server-name database-filename ...
```

适用于

所有操作系统和数据库服务器。

注释

缺省情况下，数据库服务器将接受第一个不包含路径和扩展名的数据库文件名称。例如，如果服务器通过 `samples-dir\demo.db` 文件启动，且没有指定 `-n` 选项，则服务器的名称将为 `demo`。

当某个数据库服务器启动时，它会尝试成为该计算机上的缺省数据库服务器。在没有缺省服务器时，要启动的第一个数据库服务器会成为缺省数据库服务器。该计算机上未显式指定数据库服务器名称的共享内存连接尝试将连接到缺省服务器。

注意

建议对正在由已部署的应用程序使用的数据库服务器使用 `-xd` 选项，并且所有客户端都通过使用 `ENG` 连接参数来明确指定它们应连接到的数据库服务器的名称。这样可确保当一台计算机上有多多个 SQL Anywhere 数据库服务器正在运行时，数据库能够连接到正确的数据库服务器。

不会对服务器名进行任何字符集转换。如果客户端字符集和数据库服务器字符集不同，则在服务器名中使用扩展字符可能导致找不到服务器。如果客户端和服务器分别运行不同的操作系统或区域设置，则应在服务器名中使用 7 位 ASCII 字符。请参见“[连接字符串和字符集](#)”一节第 386 页。

数据库服务器名称必须是有效的标识符。长数据库服务器名将被截断为不同的长度，具体长度取决于协议。数据库服务器名不能：

- 以空格、单引号或双引号开头
- 以空格结尾
- 含有分号
- 超过 250 个字节

注意

对于 Windows 和 Unix，9.0.2 和更早期版本的客户端不能连接到名称超过以下长度的 10.0.0 版和更高版本的数据库服务器：

- 对于 Windows 共享内存，为 40 字节
- 对于 Unix 共享内存，为 31 字节
- 对于 TCP/IP，为 40 字节

服务器名称指定要在客户端应用程序连接字符串或配置文件的 `ServerName (ENG)` 连接参数中使用的名称。对于共享内存，除非指定了 `-xd`，否则在没有指定服务器名称的情况下将使用缺省的数据库服务器，前提是该计算机上至少运行一个数据库服务器。

建议不运行多台名称相同的数据库服务器。

有两个 -n 选项

`-n` 选项与位置有关。如果此选项出现在任何数据库文件名之前，则为服务器选项并为服务器命名。如果此选项出现在数据库文件名之后，则为数据库选项并为数据库命名。

例如，以下命令为数据库服务器 `SERV` 和数据库 `DATA` 命名：

```
dbsrv11 -n SERV sales.db -n DATA
```

请参见“[-n 数据库选项](#)”一节第 239 页。

另请参见

- “标识符”一节 《SQL Anywhere 服务器 - SQL 参考》
- “`ServerName` 连接参数 [ENG]”一节第 280 页
- “命名服务器和数据库”一节第 43 页
- “`-xd` 服务器选项”一节第 223 页

-o 服务器选项

将所有数据库服务器消息输出到数据库服务器消息日志文件中。

语法

```
{ dbsrv11 | dbeng11 } -o filename ...
```

适用于

所有操作系统和数据库服务器。

注释

将所有数据库服务器消息（包括信息性消息、错误、警告以及 MESSAGE 语句输出）输出到指定文件并显示在数据库服务器消息窗口中。如果指定 `-qi` 和 `-o` 选项，则所有消息将仅显示在数据库服务器消息日志文件中。

建议不要使用 `.log` 作为文件名的结尾，因为这将导致使用事务日志执行操作的实用程序发生问题。

可通过执行以下命令获取数据库服务器消息日志文件的名称：

```
SELECT PROPERTY ( 'ConsoleLogFile' );
```

另请参见

- “记录数据库服务器操作” 一节第 40 页
- “`-oe` 服务器选项” 一节第 197 页
- “`-on` 服务器选项” 一节第 198 页
- “`-os` 服务器选项” 一节第 198 页
- “`-ot` 服务器选项” 一节第 199 页
- “`-qi` 服务器选项” 一节第 201 页

-oe 服务器选项

指定用于记录启动错误、致命错误和声明的文件的名称。

语法

```
{ dbsrv11 | dbeng11 } -oe filename ...
```

适用于

所有操作系统和数据库服务器。

注释

该输出日志文件中的每一行都以日期和时间为前缀。启动错误包括：

- 无法打开/读取数据库文件：*database file*
- 具有该名称的数据库服务器已经启动

无论是否指定 `-oe` 选项，致命错误和声明都会记录到 Windows 应用程序事件日志（Windows Mobile 除外）或 Unix 系统日志中。

建议不要使用 `.log` 作为文件名的结尾，因为这将导致使用事务日志执行操作的实用程序发生问题。

另请参见

- “`-o` 服务器选项” 一节第 196 页
- “`-on` 服务器选项” 一节第 198 页
- “`-os` 服务器选项” 一节第 198 页
- “`-ot` 服务器选项” 一节第 199 页
- “`-qi` 服务器选项” 一节第 201 页

-on 服务器选项

指定数据库服务器消息日志的最大大小，达到该大小后，将用扩展名 *.old* 重命名该文件并起用一个新文件。

语法

```
{ dbsrv11 | dbeng11 } -on { size[ k | m | g ] } ...
```

适用于

所有操作系统和数据库服务器。

注释

size 是数据库服务器消息日志文件大小的最大值，以字节为单位。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。最小值限制为 10 KB。缺省情况下无最大值限制。

当数据库服务器消息日志达到指定大小后，数据库服务器将用扩展名 *.old* 重命名文件，并以原始名称起用一个新文件。

注意

如果 *.old* 数据库服务器消息日志文件已经存在，则将覆盖它。为避免丢失旧的数据库服务器消息日志文件，请改为使用 *-os* 选项。

此选项不能与 *-os* 选项一起使用。

建议不要使用 *.log* 作为数据库服务器消息日志文件的扩展名，因为这将导致使用事务日志执行操作的实用程序发生问题。

另请参见

- “记录数据库服务器操作”一节第 40 页
- “-o 服务器选项”一节第 196 页
- “-oe 服务器选项”一节第 197 页
- “-os 服务器选项”一节第 198 页
- “-ot 服务器选项”一节第 199 页

-os 服务器选项

指定数据库服务器消息日志文件的最大大小，达到此大小时重命名该文件。

语法

```
{ dbsrv11 | dbeng11 } -os { size[ k | m | g ] } ...
```

适用于

所有操作系统和数据库服务器。

注释

size 是用于记录数据库服务器消息的文件大小的最大值，以字节为单位。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。最小值限制为 10 KB。缺省情况下无最大值限制。

在数据库服务器将输出消息记录到数据库服务器消息日志文件之前，它会检查当前文件的大小。如果日志消息会导致文件大小超过指定值，数据库服务器会将数据库服务器消息日志文件重命名为 *yymmddxx.slg*，其中 *yymmdd* 代表文件创建的年、月和日，*xx* 是从 00 开始并连续递增的数字。

您可以通过该选项标识能够删除以释放磁盘空间的旧数据库服务器消息日志文件。

此选项不能与 `-on` 选项一起使用。

建议不要使用 *.log* 作为数据库服务器消息日志文件的扩展名，因为这将导致使用事务日志执行操作的实用程序发生问题。

另请参见

- “记录数据库服务器操作” 一节第 40 页
- “`-o` 服务器选项” 一节第 196 页
- “`-oe` 服务器选项” 一节第 197 页
- “`-on` 服务器选项” 一节第 198 页
- “`-ot` 服务器选项” 一节第 199 页

-ot 服务器选项

截断数据库服务器消息日志文件并将输出消息附加到该文件。

语法

```
{ dbsrv11 | dbeng11 } -ot logfile ...
```

适用于

所有操作系统和数据库服务器。

注释

其功能与 `-o` 选项相同，只是在向数据库服务器消息日志文件中写入任何消息之前，该日志文件会被截断。可使用以下命令获取数据库服务器消息日志文件的名称：

```
SELECT PROPERTY ( 'ConsoleLogFile' );
```

建议不要使用 *.log* 作为数据库服务器消息日志文件的扩展名，因为这将导致使用事务日志执行操作的实用程序发生问题。

另请参见

- “记录数据库服务器操作” 一节第 40 页
- “`-o` 服务器选项” 一节第 196 页
- “`-oe` 服务器选项” 一节第 197 页
- “`-on` 服务器选项” 一节第 198 页
- “`-os` 服务器选项” 一节第 198 页

-p 服务器选项

设置通信包的最大大小。

语法

```
{ dbsrv11 | dbeng11 } -p integer ...
```

适用于

所有操作系统和数据库服务器。

注释

在所有操作系统（Windows Mobile 除外）上，缺省值为 7300 个字节。在 Windows Mobile 上，缺省值为 1460 个字节。最小值为 500 个字节，最大值为 16000 个字节。

您可以通过设置 CommBufferSize (CBSIZE) 连接参数来更改连接的通信缓冲区大小。

另请参见

- [“-pc 服务器选项”一节第 200 页](#)
- [“-pt 服务器选项”一节第 201 页](#)
- [“CommBufferSize 连接参数 \[CBSIZE\]”一节第 253 页](#)

-pc 服务器选项

压缩除同机连接之外的所有连接。

语法

```
dbsrv11 -pc ...
```

适用于

所有操作系统和网络服务器，Web 服务器除外。

注释

使用 -pc 选项可压缩在 SQL Anywhere 客户端和服务器之间发送的包。在某些情况下，压缩连接可以提高性能。含有高度可压缩数据的大数据量传输通常可获得最佳的压缩率。通过在客户端的连接参数中指定 COMPRESS=NO，可以为特定的客户端替换此选项。

缺省情况下，不压缩连接。如果指定 -pc 选项，将会压缩除同机连接、Web 服务连接和 TDS 连接之外的所有连接。TDS 连接（包括 jConnect）不支持 SQL Anywhere 通信压缩。

通过任何通信链接进行的同一台计算机的连接不会压缩，即使是使用 -pc 选项或 COMPRESS=YES 连接参数也是如此。

另请参见

- “-p 服务器选项” 一节第 200 页
- “-pt 服务器选项” 一节第 201 页
- “调整通信压缩设置以改善性能” 一节第 141 页
- “Compress 连接参数 [COMP]” 一节第 255 页
- “使用压缩功能” 一节 《SQL Anywhere 服务器 - SQL 的用法》

-pt 服务器选项

增加或减少压缩包的大小限制。

语法

```
dbsrv11 -pt size ...
```

适用于

所有操作系统和网络服务器。

注释

此参数采用一个整数值来表示要压缩的包的最小字节数。建议不要使用小于 80 的值。缺省值为 120 个字节。

在有些情况下，更改压缩阈值可使得压缩包操作只有在压缩能提高包的传输速度时才进行，从而提高压缩连接的性能。缺省设置应适用于大多数情况。

如果客户端和服务器指定了不同的压缩阈值设置，则使用客户端设置。

另请参见

- “-p 服务器选项” 一节第 200 页
- “-pc 服务器选项” 一节第 200 页
- “调整通信压缩设置以改善性能” 一节第 141 页
- “CompressionThreshold 连接参数 [COMPTH]” 一节第 256 页
- “使用压缩功能” 一节 《SQL Anywhere 服务器 - SQL 的用法》

-qi 服务器选项

控制是否显示数据库服务器系统托盘图标和数据库服务器消息窗口。

语法

```
{ dbsrv11 | dbeng11 } -qi ...
```

适用于

Windows

注释

除了可能出现的启动错误窗口外，此选项不留下显示服务器正在运行的任何迹象。可以使用 `-o` 和/或 `-oe` 日志文件来诊断错误。

另请参见

- “`-qn` 服务器选项” 一节第 202 页
- “`-qp` 服务器选项” 一节第 203 页
- “`-qs` 服务器选项” 一节第 203 页
- “`-qw` 服务器选项” 一节第 204 页
- “`-o` 服务器选项” 一节第 196 页
- “`-oe` 服务器选项” 一节第 197 页

-qn 服务器选项

指定在启动时不最小化数据库服务器消息窗口。

语法

```
{ dbsrv11 | dbeng11 } -qn ...
```

适用于

Windows

Linux（如果使用 X 窗口服务器）

注释

缺省情况下，数据库服务器消息窗口在数据库服务器启动完成后会自动最小化。当指定该选项时，数据库服务器消息窗口在数据库服务器启动后不会最小化。

如果自动启动数据库服务器的应用程序未处于活动状态且指定了 `-qn`，则数据库服务器消息窗口可能在后台中出现。

在 Linux 上，必须在指定 `-qn` 选项的同时指定 `-ux` 选项（使用 X 窗口服务器）。

另请参见

- “`-ux` 服务器选项” 一节第 219 页
- “`-qi` 服务器选项” 一节第 201 页
- “`-qp` 服务器选项” 一节第 203 页
- “`-qs` 服务器选项” 一节第 203 页
- “`-qw` 服务器选项” 一节第 204 页

示例

以下命令将在 Linux 或 Solaris 上启动数据库服务器，显示数据库服务器消息窗口，且在数据库服务器启动后不最小化数据库服务器消息窗口：

```
dbeng11 -ux -qn sample.db
```

-qp 服务器选项

指定在数据库服务器消息窗口中不显示有关性能的消息。

语法

```
{ dbsrv11 | dbeng11 } -qp ...
```

适用于

所有操作系统和数据库服务器。

注释

在数据库服务器消息窗口中不显示有关性能的消息。被取消的消息包括：

- 表 '*table-name*' 没有唯一索引或主键
- 数据库文件 "*mydatabase.db*" 包含 *nnn* 个碎片

另请参见

- [“-qi 服务器选项”一节第 201 页](#)
- [“-qn 服务器选项”一节第 202 页](#)
- [“-qs 服务器选项”一节第 203 页](#)
- [“-qw 服务器选项”一节第 204 页](#)

-qs 服务器选项

取消启动错误窗口。

语法

```
{ dbsrv11 | dbeng11 } -qs ...
```

适用于

Windows

注释

此选项会取消启动错误窗口。启动错误的示例包括：数据库服务器无法打开或读取数据库文件，或者因为具有指定名称的另一个数据库服务器正在运行而无法启动数据库服务器。

在 Windows 平台上，如果服务器不是被自动启动，这些错误会显示在窗口中，并且必须在服务器停止前清除。如果使用 -qs 选项，则不会显示这些窗口。

如果在命令行上指定了 -qs 且其不采用 @data 的形式，则在装载语言 DLL 时会出错，将不显示任何窗口。该错误不会记录到 -o 或 -oe 日志中，而是记录到 Windows 应用程序事件日志中（Windows Mobile 除外）。

如果在命令行而不是 @data 扩展中指定 -qs，则不会显示用法错误。

另请参见

- “-qi 服务器选项” 一节第 201 页
- “-qn 服务器选项” 一节第 202 页
- “-qp 服务器选项” 一节第 203 页
- “-qw 服务器选项” 一节第 204 页
- “-o 服务器选项” 一节第 196 页
- “-oe 服务器选项” 一节第 197 页

-qw 服务器选项

指定不显示数据库服务器消息窗口。

语法

```
{ dbsrv11 | dbeng11 } -qw ...
```

适用于

所有操作系统和数据库服务器。

注释

此选项取消数据库服务器消息窗口。在 Windows 平台上，数据库服务器系统托盘图标仍然可见。可以使用 -o 和/或 -oe 日志文件来诊断错误。

另请参见

- “-qi 服务器选项” 一节第 201 页
- “-qn 服务器选项” 一节第 202 页
- “-qp 服务器选项” 一节第 203 页
- “-qs 服务器选项” 一节第 203 页

-r 服务器选项

将在数据库服务器上启动的所有数据库强制为只读。不允许对数据库进行更改：数据库服务器不修改数据库文件和事务日志文件。

语法

```
{ dbsrv11 | dbeng11 } -r ...
```

适用于

所有操作系统和数据库服务器。

注释

如果在命令行上的任何数据库名之前指定该选项，则所有数据库文件（临时文件除外）都将以只读方式打开。如果在数据库名之后指定 -r 选项，则只有该特定数据库是只读的。您可以更改临时表，但 ROLLBACK 将不起作用，因为事务和回退日志已被禁用。

通过 CD-ROM 设备发布的数据库便是不能修改的数据库文件的一种示例。可使用只读模式访问此类数据库。

例如，如果您要用 INSERT 或 DELETE 语句修改数据库，则会返回 SQLSTATE_READ_ONLY_DATABASE 错误。

不能以只读模式启动需要恢复的数据库。例如，对于使用联机备份创建的数据库文件，如果启动备份时有任何打开的事务，则不能以只读模式启动这些数据库文件，因为备份副本启动时，可能需要恢复这些事务。

不能以只读模式启动开启了审计的数据库。

如果要检查备份副本的有效性，应当在只读模式下运行数据库，以使其不能用任何方式进行修改。请参见“[校验数据库](#)”一节第 859 页。

另请参见

- “[-r 数据库选项](#)”一节第 240 页
- “[auditing 选项 \[数据库\]](#)”一节第 484 页
- “[在只读介质上部署数据库](#)”一节 《[SQL Anywhere 服务器 - 编程](#)》
- “[在特殊模式下运行](#)”一节第 46 页

示例

以只读模式打开两个数据库

```
dbeng11 -r database1.db database2.db
```

仅以只读模式打开两个数据库中的第一个。

```
dbeng11 database1.db -r database2.db
```

-s 服务器选项

设置 Syslog 消息的用户 ID。

语法

```
{ dbsrv11 | dbeng11 } -s { none | user | daemon | localn } ...
```

适用于

Unix、Mac OS X

注释

设置 Syslog 工具的消息中使用的系统用户 ID。在前台启动的数据库服务器的缺省用户 ID 为 user，而在后台运行的数据库服务器（例如，由 dbspawn 启动、由客户端自动启动或使用 -ud 数据库服务器选项启动的数据库服务器）的缺省用户 ID 为 daemon。

值 none 将阻止记录任何 Syslog 消息。localn 参数允许使用设施标识符将消息重定向到文件。可以为 n 指定 0 到 7 之间的数字（包括 0 和 7）。有关详细信息，请参见 Unix Syslog(3) 手册页。

以下步骤说明如何在 Solaris 上重定向消息，也可以在 Linux、AIX 和 Mac OS X 上进行此操作。注意，在其它平台上（例如 HP-UX），*syslog.conf* 文件会位于不同的位置。可以将 */var/adm/sqlanywhere* 文件放在您希望的任何位置上。

◆ 使用设施标识符将消息重定向到文件

1. 选择一个尚未被系统中其它正在运行的应用程序使用的唯一设施标识符。

为此，您可以通过查看 */etc/syslog.conf* 文件以确定是否有任何 *localn* 设施被引用。

2. 编辑 */etc/syslog.conf* 文件并添加以下命令行，其中 *localn* 是在第 1 步中选择的设施标识符：

```
localn.err;localn.info;localn.notice /var/adm/sqlanywhere
```

3. 创建 */var/adm/sqlanywhere* 文件：

```
touch /var/adm/sqlanywhere
```

4. 通过找到 *syslogd* 的进程 ID，将已经修改了 *syslog.conf* 文件通知给 *syslogd* 进程：

```
ps -ef | grep syslogd
```

然后执行以下命令，其中 *pid* 是 *syslogd* 的进程 ID：

```
kill -HUP pid
```

5. 使用以下命令启动 SQL Anywhere 数据库服务器，其中 *localn* 是在第 1 步中选择的设施标识符：

```
dbeng11 -s localn ...
```

现在，SQL Anywhere 数据库服务器报告给 Syslog 的任何消息都将被重定向到 */var/adm/sqlanywhere* 文件。

另请参见

- “MESSAGE 语句”一节 《SQL Anywhere 服务器 - SQL 参考》

-sb 服务器选项

指定服务器对广播的响应方式。

语法

```
{ dbsrv11 | dbeng11 } -sb { 0 | 1 } ...
```

适用于

TCP/IP

注释

使用 *-sb 0* 将导致服务器不启动任何 UDP 广播监听器。除强制客户端使用 *DoBroadcast=NONE* 和 *HOST=* 选项来连接服务器外，该选项还将导致在使用 *dblocate* 时该服务器不会被列出。

使用 *-sb 1* 将导致服务器不响应来自 *dblocate* 的广播，同时保持连接逻辑不受影响。可以通过指定 *LINKS=tcip* 和 *ENG=name* 来连接服务器。

另请参见

- “BroadcastListener 协议选项 [BLISTENER]” 一节第 286 页

-sf 服务器选项

启用和禁用正在当前数据库服务器上运行的数据库的功能。

语法

```
{ dbsrv11 | dbeng11 } -sf feature-list ...
```

适用于

所有操作系统和数据库服务器。

注释

通过该选项可启用和禁用数据库服务器的功能。这些设置将影响所有运行于数据库服务器上的数据库。通过将 `secure_feature_key` 选项设置为由 `-sk` 选项指定的密钥，可为连接启用所有禁用（受保护）的功能。将 `secure_feature_key` 选项设置为由 `-sk` 指定的密钥的任何连接还可使用 `sa_server_option` 系统过程的 **SecureFeatures** 属性更改数据库服务器的受保护功能集。

`feature-list` 是针对数据库服务器的、要加以保护的功能名称或功能集逗号分隔列表。使用 `feature-name` 指示应禁用该功能，`-feature-name` 指示应从禁用功能列表中删除该功能。例如，以下命令指示仅启用 `dbspace` 功能：

```
dbeng11 -n secure_server -sf all,-dbspace
```

支持以下 `feature-name` 值（括号中的值是功能名的简写形式，也可以用这种形式指定功能名）：

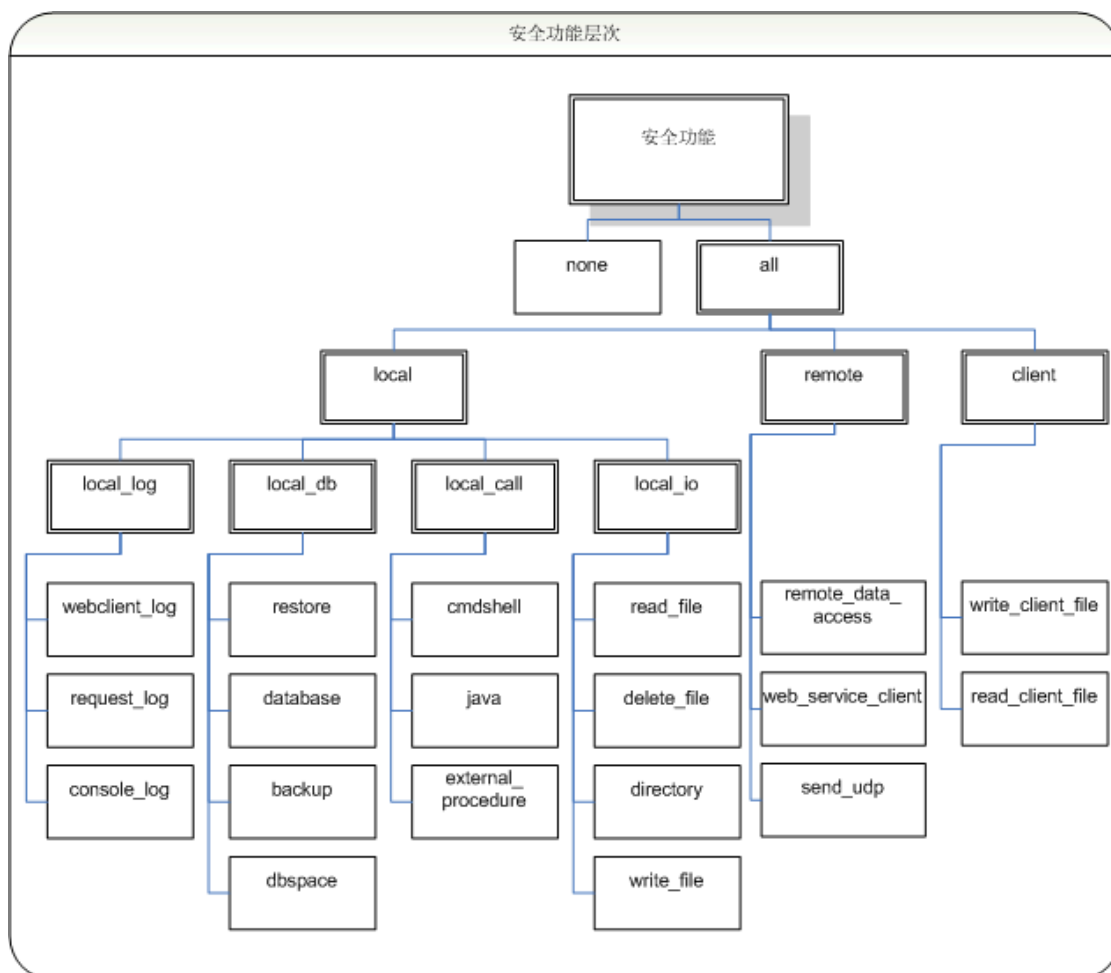
- **none** 指定未禁用任何功能。
- **all** 禁用全部可禁用的功能，包括以下组。
 - **client** 禁用所有允许访问客户端相关输入/输出的功能。这包括访问客户端计算环境。该功能集由以下功能组成。
 - **read_client_file** 禁用能够读取客户端文件的语句。例如，`READ_CLIENT_FILE` 函数和 `LOAD TABLE` 语句。请参见“访问客户端计算机上的数据”一节《SQL Anywhere 服务器 - SQL 的用法》。
 - **write_client_file** 禁用能够写入客户端文件的所有语句。例如，`UNLOAD` 语句和 `WRITE_CLIENT_FILE` 函数。请参见“访问客户端计算机上的数据”一节《SQL Anywhere 服务器 - SQL 的用法》。
 - **local** 禁用所有本地相关的功能。这包括访问服务器计算环境。该功能集由 **local_call**、**local_db**、**local_io** 和 **local_log** 功能子集组成，如下所述。
 - **local_call** 禁用所有能够执行不直接属于服务器且不受该服务器控制的代码的功能。该功能集由以下功能组成。
 - **cmdshell** 禁用 `xp_cmdshell` 过程。请参见“xp_cmdshell 系统过程”一节《SQL Anywhere 服务器 - SQL 参考》。

- **external_procedure** 禁用外部存储过程。此设置不会禁用内置于数据库服务器中的 xp_* 系统过程（例如 xp_cmdshell、xp_readfile 等）。为这些系统过程提供了单独的功能控制选项。请参见“从过程调用外部库”一节《SQL Anywhere 服务器 - 编程》。
- **java** 禁用 Java 相关的过程，例如，Java 过程。请参见“创建 Java 类以与 SQL Anywhere 配合使用”一节《SQL Anywhere 服务器 - 编程》。
- **local_db** 禁用所有与数据库文件相关的功能。该功能集由以下功能组成。
 - **backup** 禁用 BACKUP 语句，从而禁用了执行服务器端备份的功能。还可使用 dbbackup 来执行客户端备份。请参见“BACKUP 语句”一节《SQL Anywhere 服务器 - SQL 参考》。
 - **restore** 禁用 RESTORE DATABASE 语句。请参见“RESTORE DATABASE 语句”一节《SQL Anywhere 服务器 - SQL 参考》。
 - **database** 禁用 CREATE DATABASE、ALTER DATABASE、DROP DATABASE、CREATE ENCRYPTED FILE、CREATE DECRYPTED FILE、CREATE ENCRYPTED DATABASE 和 CREATE DECRYPTED DATABASE 语句。
 - **dbspace** 禁用 CREATE DBSPACE、ALTER DBSPACE 和 DROP DBSPACE 语句。
- **local_io** 禁用所有允许直接访问文件及其内容的功能。该功能集由以下功能组成。
 - **read_file** 禁用能够读取本地文件的语句。例如，xp_read_file 系统过程、LOAD TABLE 语句以及 OPENSTRING(FILE ...) 的使用。). 不建议使用替代名 load_table 和 xp_read_file。
 - **write_file** 禁用能够写入本地文件的所有语句。例如，UNLOAD 语句和 xp_write_file 系统过程。不建议使用替代名 unload_table 和 xp_write_file。
 - **delete_file** 禁用能够删除本地文件的所有语句。例如，禁用 db_delete_file DBLib 函数，该函数会删除数据库文件。db_delete_file 函数由 dbbackup -x 和 -xo 选项使用，因此在指定 -x 或 -xo 选项时保护 db_delete_file 会导致 dbbackup 失败。请参见“db_delete_file 函数”一节《SQL Anywhere 服务器 - 编程》。
 - **directory** 禁用目录类代理表。该功能在禁用 remote_data_access 时也将被禁用。
- **local_log** 禁用所有导致创建或直接向磁盘上某个文件写入数据的记录功能。该功能集由以下功能组成。
 - **request_log** 禁用更改请求日志文件名的功能，同时禁用增加请求日志文件大小或文件数量限制的功能。可在启动数据库服务器的命令中指定请求日志文件以及对此文件的限制；但在服务器启动之后将无法进行更改。禁用请求日志功能时，仍可打开和关闭请求记录功能，并可减少请求记录文件的最大文件大小和数量。请参见“请求记录”一节《SQL Anywhere 服务器 - SQL 的用法》。
 - **console_log** 禁用使用 sa_server_option 系统过程的 ConsoleLogFile 选项更改数据库服务器消息日志文件名的功能。同时禁用使用 sa_server_option 系统过程的 ConsoleLogMaxSize 选项增加日志文件最大大小的功能。可在启动数据库服务器时指定服务器日志文件及其大小。

- **webclient_log** 禁用使用 sa_server_option 系统过程的 WebClientLogFile 选项更改 Web 服务客户端日志文件名的功能。可在启动数据库服务器时指定 Web 服务客户端日志文件。请参见“-zoc 服务器选项”一节第 229 页。
- **remote** 禁用所有允许远程访问或与远程进程通信的功能。该功能集由以下功能组成。
 - **remote_data_access** 禁用任何远程数据访问服务，例如代理表。
 - **send_udp** 禁用使用 sa_send_udp 系统过程向指定地址发送 UDP 包的功能。
 - **web_service_client** 禁用 Web 服务客户端存储过程调用（即，发出 HTTP 请求的存储过程）。

功能集层次

下表列出了所有功能集关键字以及所在层次。例如，**local_io** 包括 **read_file**、**write_file**、**delete_file** 和 **directory** 功能。



另请参见

- “-sk 服务器选项” 一节第 210 页
- “secure_feature_key [数据库]” 一节第 544 页
- “sa_server_option 系统过程” 一节 《SQL Anywhere 服务器 - SQL 参考》
- “指定受保护的功能” 一节第 1010 页

示例

以下命令启动一个名为 `secure_server` 的数据库服务器，启动时将禁用对请求日志的访问权限，并禁用所有远程数据访问功能。由 `-sk` 选项指定的密钥以后可与 `secure_feature_key` 数据库选项一起使用，以为特定连接启用这些功能。

```
dbsrv11 -n secure_server -sf request_log,remote -sk j978kls12
```

如果某个与 `secure_server` 数据库服务器上运行的数据库相连的用户将 `secure_feature_key` 选项设置为由 `-sk` 指定的值，则该连接将能够访问请求日志并可使用远程数据访问功能：

```
SET TEMPORARY OPTION secure_feature_key = 'j978kls12';
```

以下命令将禁用所有功能，但本地数据库功能除外：

```
dbeng11 -n secure_server -sf all,-local_db
```

-sk 服务器选项

指定可用于启用对数据库服务器禁用的功能的密钥。

语法

```
{ dbsrv11 | dbeng11 } -sk key ...
```

适用于

所有操作系统和数据库服务器。

注释

在使用 `-sf` 选项确保数据库服务器功能的安全时，也可同时使用 `-sk` 选项，以指定可与 `secure_feature_key` 数据库选项一起使用的密钥，从而为连接启用受保护的功能。连接也可使用 `sa_server_option` 系统过程以修改对于运行于数据库服务器上的所有数据库而言均受到保护的功能或功能集。

将 `secure_feature_key` 选项的值设置为任何不同于 `-sk` 所指定的值时，将不会给出任何错误，由 `-sf` 指定的功能对该连接而言仍受到保护。

另请参见

- “-sf 服务器选项” 一节第 207 页
- “secure_feature_key [数据库]” 一节第 544 页
- “sa_server_option 系统过程” 一节 《SQL Anywhere 服务器 - SQL 参考》
- “指定受保护的功能” 一节第 1010 页

示例

以下命令将启动一个名为 `secure_server` 的数据库服务器，并禁用访问备份功能。可随后使用由 `-sk` 选项指定的密钥为特定连接启用这些功能。

```
dbsrv11 -n secure_server -sf backup -sk j978kls12
```

对于与运行于 `secure_server` 数据库服务器上的数据库的连接而言，将 `secure_feature_key` 选项设置为由 `-sk` 指定的值将允许该连接执行备份或更改在 `secure_server` 数据库服务器上禁用的功能：

```
SET TEMPORARY OPTION secure_feature_key = 'j978kls12';
```

用户然后通过执行以下命令便可为运行于 `secure_server` 上的数据库禁用所有受保护的功能：

```
CALL sa_server_option( 'SecureFeatures', 'all' );
```

-su 服务器选项

设置实用程序数据库 (`utility_db`) 的 DBA 用户的口令，或禁止与实用程序数据库连接。

语法

```
{ dbsrv11 | dbeng11 } -su password ...
```

适用于

所有操作系统和数据库服务器。

注释

该选项为实用程序数据库的 DBA 用户指定初始口令。此口令区分大小写。可将口令指定为 **none** 以禁用所有与实用程序数据库的连接。为避免在命令行上包含明文形式的实用程序数据库口令，可使用 `dbfhide` 对含有口令的文件进行模糊处理，然后在命令行上引用此经过模糊处理的文件。

如果使用个人数据库服务器且不指定 `-su` 选项，则允许使用 DBA 用户 ID 和任意口令与实用程序数据库相连。如果使用网络数据库服务器且不指定 `-su` 选项，则不允许与实用程序数据库相连，除非 `util_db.ini` 文件存在，且用户 ID 是口令与 `util_db.ini` 文件中的口令相符的 DBA。在网络服务器上，如果同时使用 `-su` 和 `util_db.ini`，则会忽略 `util_db.ini`。请注意，不建议使用 `util_db.ini` 文件。

可在连接到 `utility_db` 时执行 `CREATE USER DBA IDENTIFIED BY new-password` 语句，以更改实用程序数据库的 DBA 用户的口令。可使用 `REVOKE CONNECT FROM DBA` 语句禁用与 `utility_db` 数据库的连接。

另请参见

- “连接到实用程序数据库”一节第 29 页
- “文件隐藏实用程序 (`dbfhide`)”一节第 724 页
- “CREATE USER 语句”一节 《SQL Anywhere 服务器 - SQL 参考》
- “REVOKE 语句”一节 《SQL Anywhere 服务器 - SQL 参考》

示例

以下命令将禁用所有与实用程序数据库的连接：

```
dbeng11 -su none c:\inventory.db
```

在以下示例中，使用 `dbfhide` 对含有实用程序数据库口令的名为 `util_db_pwd.cfg` 的文件进行模糊处理并重命名为 `util_db_pwd_hide.cfg`：

```
dbfhide util_db_pwd.cfg util_db_pwd_hide.cfg
```

然后可使用 `util_db_pwd_hide.cfg` 文件指定实用程序数据库口令：

```
dbsrv11 -su @util_db_pwd_hide.cfg -n my_server c:\inventory.db
```

-ti 服务器选项

断开不活动的连接。

语法

```
{ dbsrv11 | dbeng11 } -ti minutes ...
```

适用于

所有操作系统和数据库服务器。

注释

断开在指定的 *minutes* 时间内没有提交请求的连接。缺省值为 240（4 小时）。最大值为 32767。客户端计算机在数据库事务期间保持锁定，直至事务结束或连接断开。可提供 `-ti` 选项来断开不活动的连接，解除客户端锁定。

由于大多数连接都要通过网络链接 (TCP)，所以当与 `dbsrv11` 结合使用时，`-ti` 选项会非常有用。

`-ti` 选项对仅用于本地 TCP/IP 连接的 `dbeng11` 有用。使用 `-ti` 对与使用共享内存的本地服务器的连接没有影响。

将该值设置为零将禁止检查不活动的连接，因此不会断开任何连接。

另请参见

- “`-tl` 服务器选项” 一节第 212 页
- “`sa_server_option` 系统过程” 一节 《SQL Anywhere 服务器 - SQL 参考》
- “调整超时值” 一节第 145 页

-tl 服务器选项

设置发送活动包的周期。

语法

```
{ dbsrv11 | dbeng11 } -tl seconds ...
```

适用于

所有使用 TCP/IP 的数据库服务器。

注释

活动包通过客户端/服务器 TCP/IP 通信协议定期发送，用以确认连接的完整性。如果服务器在运行了 LivenessTimeout 这样一段时间（缺省值为 2 分钟）后未检测到连接上存在活动包，则将断开通信，并且服务器会删除与该客户端相关的连接。Unix 非线性客户端和 TDS 连接不进行活动检查。

服务器上的 -tl 选项为没有指定活动周期的所有客户端设置 LivenessTimeout 值。

如果连接在 LivenessTimeout 值的三分之一到三分之二这么长的时间段内未发送任何数据包，则将发送活动数据包。

当连接超过 200 个时，服务器会根据指定的 LivenessTimeout 值自动计算一个更大的 LivenessTimeout 值，这样服务器可以更有效地处理大量连接。活动包在 LivenessTimeout 值的三分之一到三分之二这么长的一段时间内对每个空闲连接发送。不会同时发送大量活动包。如果发送活动包需要很长时间（取决于网络、计算机的硬件及计算机上的 CPU 和网络负载），则可能会在经过 LivenessTimeout 值的三分之二这么长的时间后发送活动包。如果发送活动包所用的时间很长，数据库服务器消息日志中会出现一个警告。如果出现这样的警告，请考虑增大 LivenessTimeout 的值。

尽管通常不建议这样做，但您可以指定以下语句，以禁止发送活动包：

```
dbsrv11 -tl 0
```

请考虑按如下所示的方法将 LivenessTimeout 值增大到 1 小时，而不要禁用 LivenessTimeout 选项：

```
dbsrv11 -tl 3600
```

另请参见

- “-ti 服务器选项”一节第 212 页
- “sa_server_option 系统过程”一节 《SQL Anywhere 服务器 - SQL 参考》
- “调整超时值”一节第 145 页

-tmf 服务器选项

在异常情况下，帮助从分布式事务进行恢复。

语法

```
{ dbsrv11 | dbeng11 } -tmf ...
```

适用于

Windows

注释

当分布式事务协调器不可用时，在分布式事务的恢复过程中使用。在分布式事务协调器不可用的平台上，如果在事务日志中含有分布式事务的情况下启动数据库，也可以使用该选项。

小心

如果使用此选项，则不能正确恢复分布式事务。该选项不用于日常使用。

另请参见

- [“-tmt 服务器选项”一节第 214 页](#)
- [“从分布式事务恢复”一节 《SQL Anywhere 服务器 - 编程》](#)

-tmt 服务器选项

为参与分布式事务设置重新征用超时。

语法

```
{ dbsrv11 | dbeng11 } -tmt milliseconds ...
```

适用于

Windows

注释

在分布式事务恢复期间使用。该值指定数据库服务器应等待的重新征用时间。缺省情况下，没有超时时间（数据库服务器无限期等待）。

另请参见

- [“-tmf 服务器选项”一节第 213 页](#)
- [“从分布式事务恢复”一节 《SQL Anywhere 服务器 - 编程》](#)

-tq 服务器选项

在指定时间关闭服务器。

语法

```
{ dbsrv11 | dbeng11 } -tq { datetime | time } ...
```

适用于

所有操作系统和数据库服务器。

注释

此选项对设置自动脱机备份过程非常有用。请参见 [“备份和数据恢复”第 817 页](#)。

该时间格式为 *hh:mm*（24 小时制），并且可以放在可选日期之后。如果指定了日期，则日期和时间必须括在双引号中，并且使用 *YYYY/MM/DD HH:MM* 格式。

另请参见

- [“sa_server_option 系统过程”一节 《SQL Anywhere 服务器 - SQL 参考》](#)

-u 服务器选项

使用操作系统磁盘高速缓存打开文件。

语法

```
{ dbsrv11 | dbeng11 } -u ...
```

适用于

Windows、Unix

注释

除数据库高速缓存外，还可以使用操作系统磁盘高速缓存来打开文件。

虽然操作系统磁盘高速缓存在有些情况下会改善性能，但通常不使用此选项，而是只使用数据库高速缓存，这样可以获得更好的性能。

如果服务器运行在专用计算机上，则不应使用 `-u` 选项，因为通常数据库高速缓存本身更为高效。如果服务器在有多个其它应用程序的计算机上运行（因此，大的数据库高速缓存可能会干扰其它应用程序），而且 IO 密集型任务在服务器上经常被执行（这样，大的高速缓存可提高性能），则可以考虑使用 `-u` 选项。

-ua 服务器选项

禁用异步 I/O。

语法

```
{ dbsrv11 | dbeng11 } -ua ...
```

适用于

Linux

注释

缺省情况下，在 Linux 中，数据库服务器在可能的情况下会使用异步 I/O。要使用异步 I/O，必须满足以下条件：

1. `libaio.so` 库可以在运行时装载。
2. 内核支持异步 I/O。

如果想要禁用异步 I/O，请在数据库服务器的命令行上指定 `-ua` 选项。

-uc 服务器选项

以 shell 模式启动数据库服务器。这是缺省设置。

语法

{ **dbsrv11** | **dbeng11** } **-uc** ...

适用于

Unix、Mac OS X

注释

以 shell 模式启动数据库服务器。您应只指定 **-uc**、**-ui**、**-um** 或 **-ux** 中的一个。若指定 **-uc**，则启动数据库服务器的方式与该软件的先前版本相同。

有关将数据库服务器作为守护程序启动的详细信息，请参见“**-ud 服务器选项**”一节第 216 页。

另请参见

- “**-ui 服务器选项**”一节第 217 页
- “**-um 服务器选项**”一节第 218 页
- “**-ux 服务器选项**”一节第 219 页

-ud 服务器选项

作为守护程序运行。

语法

{ **dbsrv11** | **dbeng11** } **-ud** ...

适用于

Unix、Mac OS X

注释

使用该选项您可以让服务器在当前用户会话结束之后继续运行。

使用 **-ud** 选项直接启动守护程序时，在守护程序将自身初始化或尝试打开命令中指定的任一数据库之前，**dbeng11** 和 **dbsrv11** 命令会创建守护程序进程并立即返回（退出并允许下一个命令得到执行）。

使用 **dbspawn** 而不是 **-ud** 选项的一个好处是，在确认守护程序已经启动且准备好接受请求之前，**dbspawn** 进程不会关闭。如果守护程序由于某种原因而无法启动，则 **dbspawn** 的退出代码将不为零。

另请参见

- “**后台启动服务器实用程序 (dbspawn)**”一节第 781 页
- “**软件组件的退出代码**”一节 《SQL Anywhere 服务器 - 编程》
- “**在当前会话外部运行服务器**”一节第 57 页
- “**安全提示**”一节第 1005 页

-uf 服务器选项

指定在发生致命错误时要采取的操作。

语法

```
{ dbsrv11 | dbeng11 } -uf action ...
```

适用于

Unix、Mac OS X

注释

使用该选项指定在发生致命错误时将采取以下哪种操作：

- **abort** 调用 Unix 中止功能，并生成核心文件。
- **default** 在所有情况下，数据库服务器的处理方式都与 **abort** 相同，但发生设备已满的致命错误的情况除外。这种情况下，其处理方式与 **defunct** 相同。此操作将防止系统尝试在已满设备上写入核心文件。这是缺省行为。
- **defunct** 数据库服务器继续运行而不会调用中止功能。对数据库服务器进行的任何新连接尝试均会收到原始致命错误的 SQL 错误。

另请参见

- “-oe 服务器选项” 一节第 197 页
- “支持实用程序 (dbsupport)” 一节第 785 页
- “SQL Anywhere 中的错误报告” 一节第 76 页
- “记录数据库服务器操作” 一节第 40 页

-ui 服务器选项

无论 X window 服务器是否启动，在 Linux 上该选项都将打开 [服务器启动选项] 窗口、显示数据库服务器消息窗口并启动数据库服务器。如果在 Mac OS X 上没有可用的显示，-ui 会在新窗口内显示数据库服务器消息并以 shell 模式启动数据库服务器。

语法

```
{ dbsrv11 | dbeng11 } -ui ...
```

适用于

支持 X window 服务器的 Linux 和 Mac OS X

注释

Linux 上的 -ui 选项允许您在启动数据库服务器时使用 [服务器启动选项] 窗口来指定服务器选项，并且在数据库服务器启动后即显示数据库服务器消息窗口。在 Mac OS X 上，服务器消息被重定向到 *DBLauncher.app* 内的新窗口。

在 Linux 上，如果在服务器命令行上只指定 `-ui` 选项，则会显示 **[服务器启动选项]** 窗口，从中可输入用于启动数据库服务器的选项。在 Mac OS X 上，必须使用 `-ui` 选项以及其它所需的选项来启动数据库服务器。

如果指定 `-ui`，数据库服务器会尝试查找一个可用显示。如果它找不到一个可用显示（例如，因为没有设置 `DISPLAY` 环境变量或是没有运行 X 窗口服务器），则数据库服务器会以 `shell` 模式启动。如果希望数据库服务器在找不到可用显示时不启动，则应指定 `-ux` 选项而非 `-ui` 选项。您应只指定 `-uc`、`-ui`、`-um` 或 `-ux` 中的一个。

有关将数据库服务器作为守护程序启动的信息，请参见 [“-ud 服务器选项”一节第 216 页](#)。

另请参见

- [“-uc 服务器选项”一节第 215 页](#)
- [“-um 服务器选项”一节第 218 页](#)
- [“-ux 服务器选项”一节第 219 页](#)

-um 服务器选项

在 *DBLauncher.app* 的新窗口内显示数据库服务器消息。

语法

```
{ dbsrv11 | dbeng11 } -um ...
```

适用于

Mac OS X

注释

`-um` 选项允许您连接到 *DBLauncher.app* 实例（如果它正在运行），并在 *DBLauncher.app* 的新窗口内显示消息。必须使用 `-um` 选项以及其它所需的选项来启动数据库服务器。服务器消息出现在此窗口中，而不是出现在 `shell` 中。关闭此窗口会关闭数据库服务器。如果无法与 *DBLauncher.app* 实例建立连接，数据库服务器就不能启动。

为了能将数据库服务器连接到 *DBLauncher.app* 实例，两者必须在相同的 Mac OS X 安全环境下运行。例如，由 `ssh` 会话启动的数据库服务器找不到由启动服务启动的 *DBLauncher.app* 实例。

有关将数据库服务器作为守护程序启动的信息，请参见 [“-ud 服务器选项”一节第 216 页](#)。

另请参见

- [“-uc 服务器选项”一节第 215 页](#)
- [“-ui 服务器选项”一节第 217 页](#)

-ut 服务器选项

触动临时文件。

语法

```
{ dbsrv11 | dbeng11 } -ut minutes ...
```

适用于

Unix、Mac OS X

注释

此选项使服务器按特定间隔触动临时文件。

-ux 服务器选项

在 Linux 上打开 [服务器启动选项] 窗口或显示数据库服务器消息窗口（使用 X 窗口服务器）。

语法

```
{ dbsrv11 | dbeng11 } -ux ...
```

适用于

支持 X 窗口服务器的 Linux

注释

使用 -ux 选项可在启动数据库服务器时做两件事情：在启动数据库服务器时使用 [服务器启动选项] 窗口来指定服务器选项，并在服务器启动后显示数据库服务器消息窗口。

如果在服务器命令行上只指定 -ux 选项，则会显示 [服务器启动选项] 窗口，您可在其中输入用于启动数据库服务器的选项。

如果指定 -ux，服务器必须能够找到一个可用显示。如果它找不到一个可用显示（例如如，因为没有设置 DISPLAY 环境变量或是没有运行 X 窗口服务器），则数据库服务器无法启动。如果您想要启动数据库服务器（即使它不能找到可用显示），请使用 -ui 选项而不是 -ux 选项。

如果您除了 -ux 选项以外还指定了其它服务器选项，则数据库服务器启动后即会显示数据库服务器消息窗口。您应只指定 -uc、-ui 或 -ux 中的一个。

有关将数据库服务器作为守护程序启动的详细信息，请参见“-ud 服务器选项”一节第 216 页。

另请参见

- “-uc 服务器选项”一节第 215 页
- “-ui 服务器选项”一节第 217 页
- “-qn 服务器选项”一节第 202 页

示例

以下命令显示 [服务器启动选项] 窗口，您可以在其中输入启动数据库服务器的选项：

```
dbeng11 -ux
```

以下命令启动数据库服务器并显示数据库服务器消息窗口：

```
dbeng11 -ux sample.db
```

-v 服务器选项

显示软件版本。

语法

```
{ dbsrv11 | dbeng11 } -v ...
```

适用于

所有操作系统和数据库服务器。

注释

在窗口中提供数据库服务器版本，然后停止。还可以通过右击数据库服务器消息窗口的标题栏并选择 [关于] 来获取软件版本。

-vss 服务器选项

启用和禁用卷影复制服务（Volume Shadow Copy Service，简称 VSS）。

语法

```
{ dbsrv11 | dbeng11 } -vss { + | - } ...
```

适用于

32 位 Microsoft Windows XP 和 32 位及 64 位的 Microsoft Windows 2003 及更高版本的操作系统。

注释

缺省情况下，如果 SQL Anywhere VSS 写入程序 (*dbvss11.exe*) 正在运行，则所有 SQL Anywhere 数据库都能使用 VSS 服务进行备份。您可以使用没有 SQL Anywhere VSS 写入程序的 VSS 对数据库进行备份。但是，您可能需要使用完整的 SQL Anywhere 恢复过程来恢复这些数据库。要使数据库服务器不参与 VSS 服务，请在启动数据库服务器时包括 **-vss-**。

另请参见

- “使用 SQL Anywhere 卷影复制服务 (VSS)” 一节第 829 页
- “用于 Windows 的服务实用程序 (dbsvc)” 一节第 773 页
- “从数据的介质故障中恢复” 一节第 840 页

示例

以下的命令会启动 *mydatabase.db* 数据库并指示数据库服务器不要参与 VSS 操作，即使 (*dbvss11.exe*) 写入程序正在运行：

```
dbsrv11 -vss- mydatabase.db
```

-x 服务器选项

指定服务器端网络通信协议。

语法 1

```
dbsrv11 -x { all | none | srv-protocols } ...
```

```
srv-protocols:
  { tcpip parmlist },...
parmlist:
  ( parm=value;...)
```

语法 2

```
dbeng11 -x { all | none | eng-protocols } ...
```

```
eng-protocols:
  { tcpip [ parmlist ] },...
parmlist:
  ( parm=value;...)
```

适用于

所有操作系统和数据库服务器。

注释

使用 -x 选项指定除了使用共享内存协议外，还使用哪些通信协议来监听客户端连接广播。

如果不指定 -x 选项，则服务器尝试使用操作系统上运行的数据库服务器所支持的所有协议（包括共享内存协议）来监听客户端连接广播。

如果指定 -x 选项并指定了一个或多个协议，则服务器会尝试使用指定的协议及共享内存协议来监听客户端连接广播。

有关保护 Unix 上共享内存连接的信息，请参见“安全提示”一节第 1005 页。

注意

如果您正在运行 Windows Mobile 并指定了 -x 选项，则服务器只尝试使用 TCP/IP 协议监听客户端连接广播，除非您另外显式请求使用其它协议。

无论为 -x 选项选择了何种设置，服务器将始终使用共享内存协议来监听连接广播。除了共享内存协议外，您还可以指定下列协议：

- **ALL** 使用此平台上的服务器所支持的所有通信协议（包括共享内存）来监听客户端的连接尝试。这是缺省设置。
- **NONE** 仅使用共享内存协议来监听客户端的连接尝试。
- **TCPIP (TCP)** 使用 TCP/IP 协议来监听客户端的连接尝试。网络服务器支持 TCP/IP 协议在所有操作系统之间进行通信，个人数据库服务器支持 TCP/IP 协议进行同一台计算机通信。

缺省情况下，数据库服务器在端口 2638 监听广播，并将这些广播重定向到合适的端口。在大多数情况下，这样可确保连接。

将 `-sb` 选项设置为 0，或关闭 `BroadcastListener` 选项 (`BroadcastListener=0`)，可替换此缺省设置，使服务器不在端口 2638 上监听。此外，如果客户端和服务器通过防火墙进行通信，则客户端必须通过指定 `DoBroadcast=None` 和 `Host=` 将数据包准确发送到服务器正在监听的端口。

请参见“[ServerPort 协议选项 \[PORT\]](#)”一节第 303 页。

对于某些协议，可提供附加参数，格式如下：

```
-x tcpip(PARM1=value1;PARM2=value2;...)
```

有关可用参数的详细信息，请参见“[网络协议选项](#)”一节第 284 页。

对于 Unix，如果提供多个参数，需要使用引号：

```
-x "tcpip(PARM1=value1;PARM2=value2;...)"
```

另请参见

- “[-xa 服务器选项](#)”一节第 222 页
- “[-xd 服务器选项](#)”一节第 223 页
- “[-xf 服务器选项](#)”一节第 223 页
- “[-xp 数据库选项](#)”一节第 244 页
- “[-xs 服务器选项](#)”一节第 224 页
- “[CommLinks 连接参数 \[LINKS\]](#)”一节第 254 页
- “[支持的网络协议](#)”一节第 134 页

示例

只允许共享内存和 TCP/IP 通信：

```
-x tcpip
```

-xa 服务器选项

为仲裁服务器指定逗号分隔形式的数据库名和验证字符串列表。

语法

```
dbsrv11 -xa auth=auth-strings;DBN=database-names
```

适用于

所有操作系统，但仅适用于网络服务器。

注释

仅当在数据库镜像系统中启动仲裁服务器时才指定此选项。

仲裁服务器的验证字符串必须与为主服务器和镜像服务器指定的验证字符串匹配。

如果验证字符串列表和数据库名列列表均只含有一个条目，该服务器将只作为一个数据库镜像系统的仲裁服务器；否则每个列表必须含有相同数量的条目。

另请参见

- “[DatabaseName 连接参数 \[DBN\]](#)” 一节第 260 页
- “[-sn 数据库选项](#)” 一节第 242 页
- “[-x 服务器选项](#)” 一节第 221 页
- “[-xf 服务器选项](#)” 一节第 223 页
- “[-xp 数据库选项](#)” 一节第 244 页
- “[-xs 服务器选项](#)” 一节第 224 页

示例

以下命令将启动名为 `arbiter` 的仲裁数据库服务器。

```
dbsrv11 -x tcpip -n arbiter -xa AUTH=abc;DBN=demo -xf c:\arbiterstate.txt
```

-xd 服务器选项

防止数据库服务器成为缺省数据库服务器。

语法

```
dbsrv11 -xd ...
```

适用于

所有操作系统，但仅适用于网络服务器。

注释

当某个数据库服务器启动时，它会尝试成为该计算机上的缺省数据库服务器。在没有缺省服务器时，要启动的第一个数据库服务器会成为缺省数据库服务器。该计算机上未显式指定数据库服务器名称的共享内存连接尝试将连接到缺省服务器。

指定此选项可防止数据库服务器成为缺省数据库服务器。如果指定了此选项，未指定数据库服务器名称的客户端在共享内存中将找不到数据库服务器。`-xd` 选项还可以防止数据库服务器使用缺省 TCP 端口。如果未指定 TCP 端口，数据库服务器将使用除 2638 端口之外的端口。

另请参见

- “[-n 服务器选项](#)” 一节第 195 页
- “[StartLine 连接参数 \[START\]](#)” 一节第 281 页
- “[-x 服务器选项](#)” 一节第 221 页

-xf 服务器选项

指定用于维护有关数据库镜像系统状态信息的文件的位置。

语法

```
dbsrv11 -xf state-file ...
```

适用于

所有操作系统，但仅适用于网络服务器。

注释

-xf 选项指定用于维护有关镜像系统状态信息的文件的位置。该选项是数据库镜像所必需的。缺省情况下，状态信息文件名为 *server-name.mirror_state*。

有关数据库镜像状态信息文件的详细信息，请参见“[状态信息文件](#)”一节第 885 页。

另请参见

- “-sn 数据库选项”一节第 242 页
- “-x 服务器选项”一节第 221 页
- “-xa 服务器选项”一节第 222 页
- “-xp 数据库选项”一节第 244 页
- “-xs 服务器选项”一节第 224 页

示例

以下命令（所有内容均在一行上输入）将启动名为 *server1* 的数据库服务器，其使用状态信息文件 *c:\server1state.txt*。

```
dbsrv11.exe -n server1 -x tcpip{DOBROADCAST=no}
-xf c:\server1state.txt mydemo.db -sn mirrordemo
-xp "partner=(ENG=server2;LINKS=tcpip(TIMEOUT=1));
AUTH=abc;arbiter=(ENG=arbsrv;LINKS=tcpip(TIMEOUT=1));
MODE=sync"
```

-xs 服务器选项

指定服务器端 Web 服务通信协议。

语法

```
{ dbeng11 | dbsrv11 } -xs { protocol,... } ...
```

```
protocol : {
  NONE
  | HTTP [ ( option=value;... ) ]
  | HTTPS [ ( option=value;... ) ]
```

HTTPS-only options:

```
FIPS={ Y | N }
IDENTITY=server-identity-filename
IDENTITY_PASSWORD=password
```

适用于

所有操作系统和数据库服务器。

注释

使用 -xs 选项指定要用哪些 Web 协议来监听请求。

如果不指定 `-xs` 选项，数据库服务器将不尝试监听 Web 请求。

如果在 `-xs` 选项中指定了一个或多个协议，服务器将尝试使用指定的协议来监听 Web 请求。

注意

如果您希望同时启动多个 Web 服务器，由于它们拥有相同的缺省端口，因此您必须更改其中一些服务器的端口。

您可以使用 HTTPS 或 FIPS 认可的 HTTPS 协议来保障传送层安全。请参见“[加密 SQL Anywhere Web 服务](#)”一节第 1046 页。

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](#)》。

无论您使用 `-xs` 选项时指定了哪些设置，服务器将始终使用共享内存协议来监听连接尝试。您可以指定以下任何一个参数：

- **option** 有关每个协议支持的 *option* 值的列表，请参见“[网络协议选项](#)”一节第 284 页。
- **HTTP** 使用 HTTP 协议来监听客户端的 Web 请求。所监听的缺省端口是 80。
- **HTTPS** 使用 HTTPS 协议来监听客户端的 Web 请求。所监听的缺省端口是 443。必须指定服务器的证书和口令方可使用 HTTPS。口令必须为 RSA 证书，因为 HTTPS 使用 RSA 加密方式。

SQL Anywhere HTTP 服务器支持使用 SSL 版本 3.0 和 TLS 版本 1.0 的 HTTPS 连接。

您可以指定 **HTTPS** 或带有 **FIPS=Y** 的 **HTTPS** 进行 FIPS 认可的 RSA 加密。FIPS 认可的 HTTPS 使用单独认可的库，但与 HTTPS 兼容。

注意

当使用 FIPS 认可的 HTTPS 时，Mozilla Firefox 浏览器可以连接。然而，多数版本的 Internet Explorer、Opera 或 Safari 浏览器均不支持 FIPS 认可的 HTTPS 所用的编码器套件—如果您要使用 FIPS 认可的 HTTPS，则这些浏览器可能无法连接。

有关实施经 FIPS 认可的算法的信息，请参见“[-fips 服务器选项](#)”一节第 176 页。

- **server-identity-filename** 服务器标识的路径和文件名。对于 HTTPS，必须使用 RSA 证书。
- **password** 服务器专用密钥的口令。在创建服务器证书时指定此口令。
- **NONE** 不监听 Web 请求。这是缺省设置。

有关可用参数的详细信息，请参见“[网络协议选项](#)”一节第 284 页。

对于 Unix，如果提供多个参数，需要使用引号：

```
-xs "HTTP(OPTION1=value1;OPTION2=value2;...)"
```

另请参见

- “-sn 数据库选项” 一节第 242 页
- “-x 服务器选项” 一节第 221 页
- “-xa 服务器选项” 一节第 222 页
- “-xf 服务器选项” 一节第 223 页
- “-xp 数据库选项” 一节第 244 页
- “SQL Anywhere Web 服务” 《SQL Anywhere 服务器 - 编程》

示例

监听端口 80 上的 HTTP Web 请求:

```
dbeng11 web.db -xs HTTP(PORT=80)
```

使用 HTTPS 监听 Web 请求:

```
dbeng11 web.db -xs  
HTTPS(FIPS=N;PORT=82;IDENTITY=eccserver.id;IDENTITY_PASSWORD=test)
```

-z 服务器选项

显示诊断通信消息和其它消息，用于故障排除。

语法

```
{ dbsrv11 | dbeng11 } -z ...
```

适用于

所有操作系统和数据库服务器。

注释

仅当跟踪问题时才应使用此选项。信息出现在数据库服务器消息窗口中。

另请参见

- “-ze 服务器选项” 一节第 226 页

-ze 服务器选项

在数据库服务器消息窗口中显示数据库服务器环境变量。

语法

```
{ dbsrv11 | dbeng11 } -ze ...
```

适用于

所有操作系统和数据库服务器，Windows Mobile 除外。

注释

若指定 `-ze` 选项，则启动时环境变量将列在数据库服务器消息窗口中。通过在启动数据库服务器时指定 `-o` 选项，您可以将数据库服务器消息窗口中的内容记录到一个文件中。

另请参见

- “SQL Anywhere 环境变量” 第 343 页
- “`-o` 服务器选项” 一节第 196 页
- “`-z` 服务器选项” 一节第 226 页

示例

以下命令将启动名为 `myserver` 的数据库服务器，并将为该服务器设置的环境变量输出到数据库服务器消息窗口和 `server-log.txt` 文件中。

```
dbeng11 -n myservers -ze -o server-log.txt
```

-zl 服务器选项

启用为服务器上数据库的各个连接而捕获最近预准备的 SQL 语句的功能。

语法

```
{ dbsrv11 | dbeng11 } -zl ...
```

适用于

所有操作系统和数据库服务器。

注释

也可使用 `RememberLastStatement` 服务器设置开启此功能。通过使用 `CONNECTION_PROPERTY` 函数的 `LastStatement` 值，您可以获取为连接最新准备的 SQL 语句。`sa_conn_activity` 存储过程使您能够为服务器上各数据库的所有当前连接获取最新准备的 SQL 语句。

当准备了语句时，将会设置 `LastStatement` 值；当删除了语句时，将会清除该值。每个连接仅会记忆一个语句字符串。

如果 `sa_conn_activity` 对某个连接报告了一个非空值，它很可能为该连接当前执行的语句。假如该语句已完成，则很有可能已将其删除并清除了属性值。如果某个应用程序准备了多条语句并保有其语句句柄，则 `LastStatement` 值不反映连接当前进行的操作。

对于存储过程调用，仅显示最外面的过程调用，不显示过程中的语句。

小心

当指定 `-zl` 或启用 `RememberLastStatement` 服务器设置时，任何用户都可以调用 `sa_conn_activity` 系统过程或获取 `LastStatement` 连接属性的值，以找到为任何其他用户最新准备的 SQL 语句。应慎用该选项，并在不需要时将其关闭。

另请参见

- LastStatement 属性：“连接属性”一节第 566 页
- “sa_conn_activity 系统过程”一节 《SQL Anywhere 服务器 - SQL 参考》
- “sa_server_option 系统过程”一节 《SQL Anywhere 服务器 - SQL 参考》

-zn 服务器选项

指定要保留的请求日志文件副本的数量。

语法

```
{ dbsrv11 | dbeng11 } -zn integer
```

适用于

所有操作系统和数据库服务器。

注释

如果在较长一段时间内启用了请求记录，则请求日志文件可能会变得很大。-zn 允许您指定要保留的请求日志文件副本的数量。只有在同时指定了 -zs 时，该选项才生效。-zs 选项允许您在原始日志文件达到指定大小时，创建一个新的日志文件并重命名原始日志文件。请参见“[-zs 服务器选项](#)”一节第 232 页。

例如，如果您将请求记录信息重定向到文件 *req.out*，并使用 -zn 选项指定五个请求日志文件副本，则服务器将按以下顺序创建文件：*req.out.1*、*req.out.2*、*req.out.3*、*req.out.4* 和 *req.out.5*。若这些文件都存在并且活动请求日志再次写满，则会发生以下情况：

- *req.out.1* 被删除
- 文件 *req.out.2* 至 *req.out.5* 被重命名为 *req.out.1* 至 *req.out.4*
- 活动日志的副本被重命名为 *req.out.5*

请求记录用 -zr 选项开启，并用 -zo 选项重定向到一个单独的文件。您也可以使用 sa_server_option 系统过程来设置请求日志的数量，其中 *nn* 指定请求日志文件副本的数量：

```
CALL sa_server_option('RequestLogNumFiles',nn);
```

另请参见

- “-zo 服务器选项”一节第 229 页
- “-zr 服务器选项”一节第 230 页
- “-zs 服务器选项”一节第 232 页
- “sa_server_option 系统过程”一节 《SQL Anywhere 服务器 - SQL 参考》
- “请求记录”一节 《SQL Anywhere 服务器 - SQL 的用法》

示例

在以下示例中（所有内容在一行上输入），将请求记录信息输出到一个名为 *mydatabase.log* 的请求日志文件中（其最大大小是 10 KB），并保留请求日志的三个副本：

```
dbeng11 "c:\my data\mydatabase.db" -zr all -zn 3  
-zs 10 -zo mydatabase.log
```

-ZO 服务器选项

将请求记录信息重定向到与常规日志文件不同的文件中。

语法

```
{ dbsrv11 | dbeng11 } -zo filename ...
```

适用于

所有操作系统和数据库服务器。

注释

使用 `-zr` 选项启用请求记录。通过指定 `-zo` 选项，您可以将输出从此文件定向到另一个文件（非常规日志文件）。

此选项还会阻止请求记录显示在数据库服务器消息窗口中。

另请参见

- “`-zn` 服务器选项” 一节第 228 页
- “`-zr` 服务器选项” 一节第 230 页
- “`-zs` 服务器选项” 一节第 232 页
- “请求记录” 一节 《SQL Anywhere 服务器 - SQL 的用法》

-ZOC 服务器选项

将 Web 服务客户端信息重定向到文件。

语法

```
{ dbsrv11 | dbeng11 } -zoc filename ...
```

适用于

所有操作系统和数据库服务器。

注释

Web 服务客户端日志文件包含 HTTP 请求以及针对出站 Web 服务客户端调用所记录的传输数据。在指定 `-zoc` 服务器选项时自动启用记录。可以使用 `sa_server_option` 系统过程启用和禁用记录到此文件：

```
CALL sa_server_option( 'WebClientLogging', 'ON' );
```

另请参见

- WebClientLogging 属性：“数据库服务器属性”一节第 588 页
- WebClientLogFile 属性：“数据库服务器属性”一节第 588 页
- “sa_server_option 系统过程”一节 《SQL Anywhere 服务器 - SQL 参考》
- “SQL Anywhere Web 服务” 《SQL Anywhere 服务器 - 编程》
- “CREATE FUNCTION 语句（Web 服务）”一节 《SQL Anywhere 服务器 - SQL 参考》
- “CREATE PROCEDURE 语句（Web 服务）”一节 《SQL Anywhere 服务器 - SQL 参考》

示例

以下命令将启动数据库服务器，使其在端口 80 上监听 HTTP Web 请求，并将出站 Web 服务客户端信息记录到文件 *clientinfo.txt*:

```
dbeng11 web.db -xs HTTP(PORT=80) -zoc clientinfo.txt
```

-zp 服务器选项

启用捕获查询优化程序最近使用的计划。

语法

```
{ dbsrv11 | dbeng11 } -zp ...
```

适用于

所有操作系统和数据库服务器。

注释

如果希望数据库服务器存储每个连接最近所使用的查询执行计划，则请包括此选项。也可使用通过 sa_server_option 系统过程设置的 RememberLastPlan 服务器设置启用此项功能。可使用 LastPlanText 连接属性查看最近使用的计划的文本。

另请参见

- LastPlanText 属性：“连接属性”一节第 566 页
- “sa_conn_activity 系统过程”一节 《SQL Anywhere 服务器 - SQL 参考》
- “sa_server_option 系统过程”一节 《SQL Anywhere 服务器 - SQL 参考》

-zr 服务器选项

启用操作的请求记录。

语法

```
{ dbsrv11 | dbeng11 } -zr { SQL | HOSTVARS | PLAN | PROCEDURES | TRIGGERS | OTHER |  
BLOCKS | REPLACE | ALL | YES | NONE | NO } ...
```

适用于

所有操作系统和数据库服务器。

注释

仅当跟踪问题时才应使用此选项。信息显示在数据库服务器消息窗口中或发送到请求日志。

-zr 的值将返回以下类型的信息：

- **SQL** 启用对以下内容的记录：
 - START DATABASE 语句
 - STOP DATABASE 语句
 - STOP ENGINE 语句
 - 语句准备和执行
 - EXECUTE IMMEDIATE 语句
 - 选项设置
 - COMMIT 语句
 - ROLLBACK 语句
 - PREPARE TO COMMIT 操作
 - 连接和断开连接
 - 事务开头
 - DROP STATEMENT 语句
 - 游标解释
 - 游标打开、关闭和恢复
 - 错误
- **PLAN** 启用执行计划的记录（简单形式）。如果启用过程 (PROCEDURES) 的记录，还将记录过程的执行计划。
- **HOSTVARS** 启用主机变量值的记录。如果指定 HOSTVARS，还将记录针对 SQL 列出的信息。
- **PROCEDURES** 启用从过程中执行的语句的记录。
- **TRIGGERS** 启用从触发器中执行的语句的记录。
- **OTHER** 启用 SQL 不包含的其它请求类型（如 FETCH 和 PREFETCH）的记录。然而，如果您指定 OTHER 但未指定 SQL，则其等效于指定 SQL+OTHER。包括 OTHER 可导致日志文件迅速增大，并可能对服务器性能造成负面影响。
- **BLOCKS** 启用显示何时在另一个连接上阻塞和解除阻塞某个连接的详细信息的记录。
- **REPLACE** 在记录开始时，使用同名的新（空）请求日志来替换现有的请求日志。否则，将打开现有的请求日志，并将新条目附加到文件的结尾。
- **ALL** 记录所有支持的信息。此设置等效于指定 SQL+PLAN+HOSTVARS+PROCEDURES+TRIGGERS+OTHER+BLOCKS。此设置可导致日志文件迅速增大，会对服务器的性能产生负面影响。
- **NO 或 NONE** 禁用记录请求日志。

一旦数据库服务器启动，您便可使用 `sa_server_option` 系统过程更改请求日志设置，以记录更多或更少的信息。请参见“[sa_server_option 系统过程](#)”一节《[SQL Anywhere 服务器 - SQL 参考](#)》。

使用以下查询可获得 RequestLogging 设置的当前值：

```
SELECT PROPERTY( 'RequestLogging' );
```

另请参见

- “-zn 服务器选项” 一节第 228 页
- “-zo 服务器选项” 一节第 229 页
- “请求记录” 一节 《SQL Anywhere 服务器 - SQL 的用法》

-zs 服务器选项

限制请求日志的大小。

语法

```
{ dbsrv11 | dbeng11 } -zs { size[ k | m | g ] } ...
```

适用于

所有操作系统和数据库服务器。

注释

请求记录用 -zr 选项开启，并用 -zo 选项重定向到一个单独的文件。您可以使用 -zs 选项来限制该文件的大小。

size 是请求日志文件大小的最大值，以字节为单位。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。

如果将 -zs 指定为 0，则请求记录文件没有大小限制，且该文件也永远不会被重命名。这是缺省值。

当请求日志文件达到 -zs 选项或 *sa_server_option* 系统过程指定的大小后，会用附加的扩展名 *.old* 重命名该文件（如果已经存在这样一个文件，则用相同名称替换现有文件）。然后，重新启动请求日志文件。

另请参见

- “-zn 服务器选项” 一节第 228 页
- “-zo 服务器选项” 一节第 229 页
- “-zr 服务器选项” 一节第 230 页
- “sa_server_option 系统过程” 一节 《SQL Anywhere 服务器 - SQL 参考》
- “请求记录” 一节 《SQL Anywhere 服务器 - SQL 的用法》

示例

以下示例说明如何使用 -zs 选项来控制日志文件的大小。假设用以下命令行启动数据库服务器：

```
dbeng11 -zr all -zs 10k -zo mydatabase.log
```

将创建一个新日志文件 *mydatabase.log*。当此文件的大小达到 10 KB 时，任何现有的 *mydatabase.old* 文件都会被删除，*mydatabase.log* 将被重命名为 *mydatabase.old*，并将起用一个新的 *mydatabase.log* 文件。每当 *mydatabase.log* 文件达到指定大小时（在本例中为 10 KB），都会重复这一过程。

-zt 服务器选项

启用对请求计时信息的记录。

语法

```
{ dbsrv11 | dbeng11 } -zt ...
```

适用于

所有操作系统和数据库服务器。

注释

一旦数据库服务器启动，您便可使用 `sa_server_option` 系统过程更改请求计时信息的记录状态。请参见“[sa_server_option 系统过程](#)”一节《[SQL Anywhere 服务器 - SQL 参考](#)》。

使用以下查询可获得 `RequestTiming` 设置的当前值：

```
SELECT PROPERTY( 'RequestTiming' );
```

另请参见

- “[sa_performance_diagnostics 系统过程](#)”一节《[SQL Anywhere 服务器 - SQL 参考](#)》
- “[sa_performance_statistics 系统过程](#)”一节《[SQL Anywhere 服务器 - SQL 参考](#)》
- “[请求记录](#)”一节《[SQL Anywhere 服务器 - SQL 的用法](#)》

数据库选项

这些选项在数据库文件之后指定，而且只应用于该数据库。

-a 数据库选项

应用指定的事务日志。-a 数据库选项必须在 *database-file* 之后指定，并且仅应用于该数据库。

语法

```
{ dbsrv11 | dbeng11 } [ server-options ] database-file -a log-filename ...
```

适用于

所有操作系统和数据库服务器。

注释

此选项用于从数据库文件的介质故障恢复。当指定此选项时，数据库服务器应用日志并随后关闭—服务器将不再继续运行。如果需要应用多个事务日志，则在使用 -a 时，您必须知道正确的应用顺序。如果使用 -ad 或 -ar 选项而不是 -a，数据库服务器将自动按正确的顺序应用多个事务日志。

启动服务器时指定高速缓存大小可缩短恢复时间。

请参见“备份和数据恢复”第 817 页。

另请参见

- “从数据的介质故障中恢复”一节第 840 页
- “通过多个事务日志恢复数据库”一节第 837 页
- “-ad 数据库选项”一节第 234 页
- “-ar 数据库选项”一节第 235 页
- “-as 数据库选项”一节第 236 页

示例

在以下示例中（所有内容均在一行输入），将日志文件 *demo.log* 应用到示例数据库的备份副本。

```
dbeng11 "c:\backup\demo.db" -a "c:\backup\demo.log"
```

-ad 数据库选项

指定包含要应用于数据库的事务日志文件的目录。-ad 数据库选项必须在 *database-file* 之后指定，并且仅应用于该数据库。

语法

```
{ dbsrv11 | dbeng11 } [ server-options ] database-file -ad log-directory ...
```

适用于

所有操作系统和数据库服务器。

注释

包括 `-ad` 选项时，将在指定的目录中查找与数据库相关联的事务日志文件。将按日志偏移顺序应用起始日志偏移大于或等于数据库文件中存储的起始日志偏移的事务日志文件。一旦应用了所有事务日志文件，数据库将停止。如果希望数据库在应用事务日志文件之后继续运行，还必须指定 `-as` 选项。

另请参见

- “从数据的介质故障中恢复”一节第 840 页
- “通过多个事务日志恢复数据库”一节第 837 页
- “`-a` 数据库选项”一节第 234 页
- “`-ar` 数据库选项”一节第 235 页
- “`-as` 数据库选项”一节第 236 页

示例

数据库服务器将 `backup` 目录中的日志文件应用于 `mysample.db` 数据库，然后在应用了日志文件后立即停止数据库。

```
dbeng11 "c:\mysample.db" -ad "c:\backup"
```

数据库服务器将 `backup` 目录中的日志文件应用于 `mysample.db` 数据库，数据库将在应用完日志文件后继续运行。

```
dbeng11 "c:\mysample.db" -ad "c:\backup" -as
```

`-ar` 数据库选项

指定应将当前事务日志所在目录中的所有事务日志文件都应用于数据库。`-ar` 数据库选项必须在 `database-file` 之后指定，并且仅应用于该数据库。

语法

```
{ dbsrv11 | dbeng11 } [ server-options ] database-file -ar ...
```

适用于

所有操作系统和数据库服务器。

注释

包括 `-ar` 选项时，数据库服务器将查找与数据库相关联且与当前事务日志位于同一目录中的事务日志文件。事务日志位置从数据库中获得。将按日志偏移顺序应用起始日志偏移大于或等于数据库中存储的起始日志偏移的事务日志文件。一旦应用了所有事务日志文件，数据库将停止。如果希望数据库在应用事务日志文件之后继续运行，还必须指定 `-as` 选项。

另请参见

- “从数据的介质故障中恢复”一节第 840 页
- “通过多个事务日志恢复数据库”一节第 837 页
- “-a 数据库选项”一节第 234 页
- “-ad 数据库选项”一节第 234 页
- “-as 数据库选项”一节第 236 页

示例

数据库服务器将事务日志文件（其位置从数据库中获得）应用于 *mysample.db* 数据库。在应用事务日志文件后数据库将继续运行。

```
dbeng11 "c:\mysample.db" -ar -as
```

-as 数据库选项

指定数据库应在应用事务日志后继续运行（与 `-ad` 或 `-ar` 结合使用）。`-as` 数据库选项必须在 *database-file* 之后指定，并且仅应用于该数据库。

语法

```
{ dsrv11 | dbeng11 } [ server-options ] database-file { -ad log-dir | -ar } -as ...
```

适用于

所有操作系统和数据库服务器。

注释

必须结合 `-ad` 或 `-ar` 选项来指定 `-as` 选项。当包括 `-as` 时，数据库将在应用了事务日志后继续运行。

另请参见

- “从数据的介质故障中恢复”一节第 840 页
- “通过多个事务日志恢复数据库”一节第 837 页
- “-a 数据库选项”一节第 234 页
- “-ad 数据库选项”一节第 234 页
- “-ar 数据库选项”一节第 235 页

示例

数据库服务器将事务日志文件应用于 *mysample.db* 数据库。在本示例中，因为指定了 `-ar`，数据库服务器将从数据库中获得事务日志的位置。数据库在应用日志文件后将继续运行。

```
dbeng11 "c:\mysample.db" -ar -as
```

数据库服务器将 *backup* 目录中的日志文件应用于 *mysample.db* 数据库。数据库在应用日志文件后将继续运行。

```
dbeng11 "c:\mysample.db" -ad "c:\backup" -as
```

-ds 数据库选项

指定数据库的 `dbspace` 所在的目录。`-ds` 数据库选项必须在 *database-file* 之后指定，并且仅应用于该数据库。

语法

```
{ dbsrv11 | dbeng11 } -ds dbspace-directory ...
```

适用于

所有操作系统和数据库服务器。

注释

指定一个 `dbspace` 目录之后，数据库服务器仅在此目录中搜索 `dbspace`。`dbspace` 的位置显示在数据库服务器消息窗口中。

如果备份中包含具有完整路径名称的 `dbspace`，则可以使用此选项启动原数据库所在计算机上的数据库的备份副本，同时原数据库保持运行。

另请参见

- “使用附加 `dbspace`” 一节第 23 页
- “`START DATABASE` 语句” 一节 《SQL Anywhere 服务器 - SQL 参考》
- “`STOP DATABASE` 语句” 一节 《SQL Anywhere 服务器 - SQL 参考》
- “`default_dbpace` 选项 [数据库]” 一节第 499 页

示例

以下示例会启动一个数据库服务器，该服务器将在 `c:\backup\Nov15` 目录中查找 `dbspace`：

```
dbeng11 c:\backup\Nov15\my.db -ds c:\backup\Nov15\
```

以下示例会启动一个数据库服务器，该服务器将在当前目录中查找 `dbspace`：

```
dbeng11 my.db -ds
```

-dh 数据库选项

可阻止在对此服务器使用服务器枚举实用程序 (`dblocate`) 时显示此数据库。`-dh` 数据库选项必须在 *database-file* 之后指定，并且仅应用于该数据库。

语法

```
{ dbsrv11 | dbeng11 } [ server-options ] database-file -dh ...
```

适用于

所有平台。

注释

当对服务器运行服务器枚举实用程序 (dblocate) 时, `-dh` 选项会使数据库变得不能检测。因此, 当 `dblocate` 与 `-d` 选项、`-dn` 选项或 `-dv` 选项一同使用时, 将不会列出数据库。

另请参见

- “服务器枚举实用程序 (dblocate)” 一节第 763 页

-ek 数据库选项

指定高度加密数据库的密钥。-ek 数据库选项必须在 *database-file* 之后指定, 并且仅应用于该数据库。

语法

```
{ dbsrv11 | dbeng11 } [ server-options ] database-file -ek key ...
```

适用于

所有操作系统和服务器。

注释

只有使用 `-ek` 选项提供密钥值才能启动加密的数据库。密钥是一个包括混合大小写、数字、字母和特殊字符的字符串。

如果您希望在窗口中键入加密密钥以避免明文显示, 请使用 `-ep` 服务器选项。请参见 “[-ep 服务器选项](#)” 一节第 173 页。

若要保证客户端应用程序与数据库服务器之间通信包的安全, 可使用 `-ec` 服务器选项和传送层安全。请参见 “[传送层安全](#)” 第 1031 页。

另请参见

- “[-ec 服务器选项](#)” 一节第 171 页
- “[-ep 服务器选项](#)” 一节第 173 页
- “[DatabaseKey 连接参数 \[DBKEY\]](#)” 一节第 259 页
- “[加密和解密数据库](#)” 一节第 1019 页

示例

以下示例将启动数据库并在命令行上指定加密密钥。

```
dsrv11 -x tcpip mydata.db -ek "Akmm9u70y"
```

-m 数据库选项

在检查点完成之后截断事务日志。-m 数据库选项必须在 *database-file* 之后指定, 并且仅应用于该数据库。

语法

```
{ dbsrv11 | dbeng11 } [ server-options ] database-file -m ...
```

适用于

所有操作系统和数据库服务器。

注释

在检查点完成之后截断事务日志，可以在关机时，也可以是按照服务器所安排的检查点的结果。此选项提供了自动限制事务日志增加的方法。检查点频率仍由 `checkpoint_time` 和 `recovery_time` 选项（或 `-gc` 和 `-gr` 数据库服务器命令行选项）控制。

在处理需要快速响应的大容量事务，且事务日志的内容与恢复或复制无关的情况下，`-m` 选项非常有用。当选择该选项时，对保存数据库文件的设备上的介质故障将没有保护。

为避免出现数据库文件碎片，建议您在使用该选项时，将事务日志存放在与数据库本身不同的单独设备或分区上。

此选项与 `-m` 服务器选项相同，但是它只适用于当前数据库或由 `database-file` 变量标识的数据库。

小心

不要对正在复制或同步的数据库使用 `-m` 选项。SQL Remote 和 MobiLink 所使用的复制和同步对事务日志信息有内在的依赖性。

另请参见

- “[-m 服务器选项](#)” 一节第 194 页
- “[事务日志](#)” 一节第 14 页
- “[事务日志实用程序 \(dblog\)](#)” 一节第 792 页

示例

以下示例启动一个名为 `silver` 的数据库服务器，并装载数据库 `salesdata.db`。完成检查点操作后，会删除事务日志内容。

```
dbsrv11 -n silver "c:\inventory details\salesdata.db" -m
```

-n 数据库选项

设置数据库的名称。`-n` 数据库选项必须在 `database-file` 之后指定，并且仅应用于该数据库。

语法

```
{ dbsrv11 | dbeng11 } [ server-options ] database-file -n string ...
```

适用于

所有操作系统和数据库服务器。

注释

数据库服务器和数据库都可以被命名。由于数据库服务器可装载多个数据库，因此需要使用数据库名称来区分不同的数据库。

在缺省情况下，数据库接受不带路径和扩展名的数据库文件名。例如，如果数据库通过 *samples-dir\demo.db* 启动，且没有指定 `-n` 选项，则数据库的名称将为 `demo`。

数据库名不能：

- 以空格、单引号或双引号开头
- 以空格结尾
- 含有分号
- 超过 250 个字节

只能使用数据库名 `utility_db` 连接 SQL Anywhere 实用程序数据库。请参见“[使用实用程序数据库](#)”一节第 28 页。

另请参见

- “命名服务器和数据库”一节第 43 页
- “`-n` 服务器选项”一节第 195 页

示例

以下示例启动一个高速缓存大小为 3 MB 的数据库服务器，装载数据库，并将此数据库命名为 `test`。因为没有指定数据库服务器名，服务器将采用第一个数据库的名称，因此服务器名也为 `test`。

```
dbsrv11 -c 3MB "c:\mydata.db" -n "test"
```

有两个 `-n` 选项

`-n` 选项与位置有关。如果此选项出现在数据库文件名之前，则为服务器选项并为服务器命名。如果此选项出现在数据库文件名之后，则为数据库选项并为数据库命名。

例如，以下命令为服务器 `SERV` 和数据库 `DATA` 命名：

```
dbsrv11 -n SERV c:\mydata.db -n DATA
```

请参见“[-n 服务器选项](#)”一节第 195 页。

-r 数据库选项

以只读模式启动指定的数据库。不允许对数据库进行更改：数据库服务器不修改数据库文件和事务日志文件。`-r` 数据库选项必须在 *database-file* 之后指定，并且仅应用于该数据库。

语法

```
{ dbsrv11 | dbeng11 } [ server-options ] database-file -r ...
```

适用于

所有操作系统和数据库服务器。

注释

如果在命令行上的任何数据库名之前指定该选项，则以只读模式打开除临时文件之外的所有数据库文件（主数据库文件、`dbspace`、事务日志和事务日志镜像）。如果在数据库名之后指定 `-r` 选项，则只有该特定数据库是只读的。您可以更改临时表，但 `ROLLBACK` 将不起作用，因为事务和回滚日志已被禁用。

通过 CD-ROM 设备发布的数据库便是不能修改的数据库文件的一种示例。可使用只读模式访问此类数据库。

例如，如果您要用 `INSERT` 或 `DELETE` 语句修改数据库，则会返回 `SQLSTATE_READ_ONLY_DATABASE` 错误。

不能以只读模式启动需要恢复的数据库。例如，对于使用联机备份创建的数据库文件，如果启动备份时有任何打开的事务，则不能以只读模式启动这些数据库文件，因为备份副本启动时，可能需要恢复这些事务。

如果启用审计功能，则无法以只读模式启动数据库。

另请参见

- “[-r 服务器选项](#)” 一节第 204 页
- “[auditing 选项 \[数据库\]](#)” 一节第 484 页

示例

以只读模式打开两个数据库

```
dbeng11 -r database1.db database2.db
```

仅以只读模式打开两个数据库中的第一个。

```
dbeng11 database1.db -r database2.db
```

-sm 数据库选项

提供可用于访问只读镜像数据库的替代数据库服务器名称。

语法

```
dbsrv11 [ server-options ] database-file -sm alternate-server-name
```

适用于

所有操作系统，但仅适用于网络服务器。

注释

只有在数据库服务器担当数据库的镜像时，`alternate-server-name` 才处于活动状态。通过使用 `-sm` 和 `-sn` 命令行选项，应用程序可始终连接到主服务器或镜像服务器上的数据库，而无需知道哪个物理服务器担当主服务器或镜像服务器。

另请参见

- “单独授权的组件”一节 《SQL Anywhere 11 - 简介》
- “配置在镜像服务器上运行的数据库的只读访问”一节第 898 页
- “-xa 服务器选项”一节第 222 页
- “-xf 服务器选项”一节第 223 页
- “-xp 数据库选项”一节第 244 页
- “START DATABASE 语句”一节 《SQL Anywhere 服务器 - SQL 参考》
- “数据库镜像简介”一节第 880 页
- “服务器枚举实用程序 (dblocate)”一节第 763 页
- ReadOnly 属性: “数据库属性”一节第 601 页

示例

以下命令将在名为 `myserver` 的数据库服务器上启动数据库 `satest.db` 和 `sample.db`。-sn 选项指示数据库服务器使用 `mysampleprimary` 作为连接到 `sample.db` 时的替代服务器名，而 -sm 选项指示数据库服务器使用 `mysamplemirror` 作为连接到在镜像服务器上运行的 `sample.db` 时的替代服务器名。

```
dbsrv11 -n myservers satest.db sample.db -sn mysampleprimary -sm
mysamplemirror
-xp "partner=( ENG=server2;LINKS=TCPIP( PORT=2637;TIMEOUT=1 ) );auth=abc;
arbiter=( ENG=arbiter;LINKS=TCPIP( PORT=2639;TIMEOUT=1 ) );mode=sync"
```

当在 *主服务器* 上运行 `sample.db` 时，可以使用以下任何连接参数连接到 `sample.db`：

- ENG=myserver;DBN=sample
- ENG=mysampleprimary
- ENG=mysampleprimary;DBN=sample

使用 ENG=mysampleprimary 无法连接到 `satest.db`。

当在 *镜像服务器* 上运行 `sample.db` 时，可以使用以下任何连接参数连接到 `sample.db`：

- ENG=myserver;DBN=sample
- ENG=mysamplemirror
- ENG=mysamplemirror;DBN=sample

使用 ENG=mysamplemirror 无法连接到 `satest.db`。

-sn 数据库选项

为运行于数据库服务器上的单个数据库提供替代服务器名。-sn 数据库选项必须在 `database-file` 之后指定，并且仅应用于该数据库。

语法

```
dbsrv11 [ server-options ] database-file -sn alternate-server-name
```

适用于

所有操作系统，但仅适用于网络服务器。

注释

可将数据库服务器配置为监听某个特定数据库服务器的多个服务器名。真实服务器名之外的其它服务器名称为替代服务器名，它们特定于在数据库服务器上运行的特定数据库。使用替代服务器名的客户端只能连接到指定了替代服务器名的数据库。

网络上的替代服务器名必须唯一；否则数据库将无法启动。如果数据库以服务器命令方式启动且替代服务器名不唯一，服务器将无法启动。也可使用 `START DATABASE` 语句提供替代服务器名。

指定替代服务器名的客户端只能连接到指定了替代服务器名的数据库。这些客户端无法连接到在该数据库服务器上运行的其它任何数据库。如果指定 `DBN` 或 `DBF` 连接参数，则必须分别与数据库名或数据库文件相匹配。如果未指定 `DBN` 或 `DBF` 连接参数，则数据库将担当该服务器的缺省数据库。

服务器枚举实用程序 (`dblocate`) 将检测替代服务器名。

将替代服务器名用于数据库镜像

使用数据库镜像时，必须为客户端应用程序指定替代服务器名，这样才能在事先不知道哪个服务器为主服务器、哪个服务器为镜像服务器的情况下，连接到当前的主服务器。这两个运行服务器必须使用相同的替代服务器名。

另请参见

- [“单独授权的组件”一节 《SQL Anywhere 11 - 简介》](#)
- [“-xa 服务器选项”一节第 222 页](#)
- [“-xf 服务器选项”一节第 223 页](#)
- [“-xp 数据库选项”一节第 244 页](#)
- [“START DATABASE 语句”一节 《SQL Anywhere 服务器 - SQL 参考》](#)
- [“数据库镜像简介”一节第 880 页](#)
- [“服务器枚举实用程序 \(dblocate\)”一节第 763 页](#)
- `AlternateServerName` 属性：[“数据库属性”一节第 601 页](#)

示例

以下命令将在名为 `myserver` 的数据库服务器上启动数据库 `satest.db` 和 `sample.db`。`-sn` 选项指示数据库服务器在连接到 `sample.db` 时使用 `mysample` 作为替代服务器名。

```
dbsrv11 -n myservers satest.db sample.db -sn mysample
```

可使用以下任何连接参数连接到 `sample.db`：

- `ENG=myserver;DBN=sample`
- `ENG=mysample`
- `ENG=mysample;DBN=sample`

使用 `ENG=mysample` 无法连接到 `satest.db`。

-xp 数据库选项

使用数据库镜像时，为运行服务器提供信息，使其能与伙伴服务器和仲裁服务器相连。-xp 数据库选项必须在 *database-file* 之后指定，并且仅应用于该数据库。

语法

```
dbsrv11 [ server-options ] database-file
-xp partner=( partner-conn );
auth=auth-str;
[ ;arbiter=( arbiter-conn ) ]
[ ;mode=[ sync | async | page ]
[ ;autofailover=[ YES | NO ] ]
[ ;pagetimeout=n ]
[ ;preferred=[ YES | NO ] ...
```

适用于

除 Windows Mobile 之外的所有操作系统，仅适用于网络服务器。

注释

指定 -xp 时，还必须使用 -xf 选项指定数据库镜像状态信息文件的位置。

如果在 -xp 选项中指定的连接参数无效，并且在服务器上运行多个数据库，则镜像数据库启动失败并且不尝试重新连接。如果镜像数据库是在数据库服务器上运行的唯一数据库，则数据库服务器不会启动。

partner-conn 指定伙伴服务器的连接字符串。不需要用户 ID 和口令。建议指定超时时间以减少故障转移时间。

auth-str 指定仲裁服务器所用的验证字符串。

arbiter-conn 指定用于仲裁服务器的连接字符串。不需要用户 ID 和口令。建议指定超时时间以减少故障转移时间。

mode 指定用于数据库镜像的同步模式：同步 (sync)、异步 (async) 或异步满页 (page)。

autofailover 指定镜像服务器是否在原始主服务器出现故障时自动作为主服务器进行接管。该选项不适用于同步模式。

注意

建议在使用异步模式或异步满页模式时，将 -xp autofailover 选项设置为 yes。这样，如果主服务器出现故障，镜像服务器会自动作为主服务器进行接管。

pagetimeout 指定事务日志页（无论是否已满）发往镜像服务器的频率（以秒为单位）。此选项仅在使用异步满页模式时适用。

preferred 指定该服务器是否为镜像系统中的首选服务器。只要有可能，首选服务器就会履行主服务器的角色。请参见“指定首选数据库服务器”一节第 897 页。

另请参见

- “单独授权的组件”一节 《SQL Anywhere 11 - 简介》
- “选择数据库镜像模式”一节第 883 页
- “-sn 数据库选项”一节第 242 页
- “-xa 服务器选项”一节第 222 页
- “-xf 服务器选项”一节第 223 页
- MirrorMode 属性：“数据库属性”一节第 601 页

示例

以下命令将为名为 server2 的伙伴服务器和名为 arbsrv 的仲裁服务器指定参数。

```
dbsrv11 -n server1 mydata.db -sn mydata  
-xp "partner=(ENG=server2;LINKS=tcPIP(TIMEOUT=1));  
AUTH=abc;arbiter=(ENG=arbsrv;LINKS=tcPIP(TIMEOUT=1))"
```

连接参数和网络协议选项

目录

连接参数	248
网络协议选项	284

连接参数

连接参数包括在连接字符串中。可在以下位置输入这些参数：

- 在应用程序的连接字符串中。请参见“汇集一系列连接参数”一节第 125 页和“以连接字符串形式传递的连接参数”一节第 81 页。
- 在 ODBC 数据源中。请参见“创建 ODBC 数据源”一节第 90 页。
- 在 SQL Anywhere [连接] 窗口中。请参见“通过 SQL Anywhere 实用程序连接”一节第 123 页。

在 Windows 操作系统中，[SQL Anywhere 11 的 ODBC 配置] 窗口和 SQL Anywhere [连接] 窗口共用相同的格式。有些参数与这些窗口中的复选框和字段相对应，而另外一些参数则可输入到 [高级] 选项卡上的文本框中。

注意

- 虽然连接参数的值可能区分大小写（例如 Unix 的文件名），但连接参数不区分大小写。
- 布尔参数用 YES、Y、ON、TRUE、T 或 1 打开，用 NO、N、OFF、FALSE、F 和 0 关闭。这些参数不区分大小写。
- 每个连接参数的用法将介绍该参数的使用环境。常见的用法条目包括：
 - **嵌入式数据库** 当 SQL Anywhere 用作嵌入式数据库时，连接会启动个人服务器并装载该数据库。当应用程序从该数据库断开连接时，该数据库就会被卸载，而且服务器会停止运行。
 - **运行本地数据库** 这是指 SQL Anywhere 个人服务器已在运行，且该数据库已在服务器上装载的情况。
 - **网络服务器** 当 SQL Anywhere 用作网络服务器时，客户端应用程序必须查找已在网络上的其它位置运行的服务器并连接到数据库。
- 您可以使用 dbping 实用程序来测试连接字符串。-c 选项用于指定连接参数。例如，假设名为 demo11 的个人服务器正在运行示例数据库（该数据库可使用命令 dbeng11 samples-dir \demo.db 启动）。如果名为 demo11 的数据库服务器正在本地计算机上运行并且还有一个名为 demo 的数据库正在其上运行，以下字符串将返回消息 [强制数据库回应成功]：

```
dbping -d -c "ENG=demo11;DBN=demo;UID=DBA;PWD=sql"
```

但是，如果本地计算机上未运行任何名为 other-server 的数据库服务器，以下命令会返回消息 [强制数据库回应失败 - 数据库服务器未运行]：

```
dbping -d -c "ENG=other-server;UID=DBA;PWD=sql"
```

请参见“Ping 实用程序 (dbping)”一节第 757 页。

另请参见

- “连接参数”一节第 80 页

AppInfo 连接参数 [APP]

帮助管理员识别某数据库服务器的特定的客户端连接。

用法

任何位置

值

字符串

缺省值

空字符串

注释

此连接参数既可以从嵌入式 SQL、ODBC、OLE DB 或 ADO.NET 客户端发送到数据库服务器，也可使用 iAnywhere JDBC 驱动程序从应用程序发送到数据库服务器。它在 Open Client 或 jConnect 应用程序中不可用。

它由存有客户端进程信息（如客户端计算机的 IP 地址、运行此客户端进程的操作系统等）的生成的字符串组成。该字符串在数据库服务器中与特定连接相关联，您可以使用以下语句对其进行检索：

```
SELECT CONNECTION_PROPERTY( 'AppInfo' );
```

客户端还可指定自己的字符串，该字符串被附加到生成的字符串上。AppInfo 属性字符串是一系列由分号分隔的 **key=value** 对。有效的键值如下：

- **API** DBLIB、ODBC、OLEDB、ADO.NET、iAnywhereJDBC、PHP、PerlDBD 或 DBEXPRESS。
- **APPINFO** 如果在连接字符串中指定了 AppInfo，则输入该字符串。
- **EXE** 客户端可执行文件的名称（Windows、Linux 和 Solaris）。
- **HOST** 客户端计算机的主机名称。
- **IP** 客户端计算机的 IP 地址。
- **OS** 操作系统名和版本号（例如，Windows 2000）。
- **OSUSER** 与客户端进程相关联的操作系统用户名。如果客户端进程模拟另一个用户（或者设置的 ID 位在 Unix 上设置），则返回被模拟的用户名。对于 10.0.1 版及更早版本的客户端，以及 HTTP 和 TDS 客户端，将返回空字符串。
- **PID** 客户端的进程 ID（仅限 Windows 和 Unix）。
- **THREAD** 客户端的线程 ID（仅限 Windows 和 Unix）。
- **TIMEZONEADJUSTMENT** 必须添加到协调通用时间 (UTC) 中以显示连接本地时间的分钟数。
- **VERSION** 正在使用的客户端库的版本，包括主版本号、子版本号和内部版本号（例如，11.0.0.2023）。

如果您在客户端连接参数中指定调试日志文件，则 APPINFO 字符串会添加到该文件中。

另请参见

- “连接参数”一节第 80 页
- “解决连接参数冲突”一节第 81 页
- “request_timeout 选项 [数据库]”一节第 540 页
- AppInfo 属性：“连接属性”一节第 566 页

示例

从 Interactive SQL 连接到示例数据库（缺省情况下使用 iAnywhere JDBC 驱动程序）：

```
dbisql -c "UID=DBA;PWD=sql;DBF=samples-dir\demo.db"
```

查看应用程序信息：

```
SELECT CONNECTION_PROPERTY( 'AppInfo' );
```

结果如下所示（在单个字符串中）：

```
IP=ip-address;
HOST=computer-name;
OSUSER=user-name;
OS='Windows XP Build 2600 Service Pack 2';
EXE='C:\Program Files\SQL Anywhere 11\Bin32\dbisql.exe';P
ID=0xcac;
THREAD=0xca8;VERSION=11.0.0.1200;
API=iAnywhereJDBC;
TIMEZONEADJUSTMENT=-240
```

从 Interactive SQL 连接到示例数据库，并将自己的信息附加到 AppInfo 属性：

```
dbisql -c "UID=DBA;PWD=sql;DBF=samples-dir\demo.db;APP=Interactive SQL
connection"
```

查看应用程序信息：

```
SELECT CONNECTION_PROPERTY( 'AppInfo' );
```

结果如下所示（在单个字符串中）：

```
IP=ip-address;
HOST=computer-name;
OSUSER=user-name;
OS=Windows XP Build 2600 Service Pack 2;
EXE=C:\Program Files\SQL Anywhere 11\Bin32\dbisql.exe;
PID=0xcac;
THREAD=0xba8;
VERSION=11.0.0.1200;
API=iAnywhereJDBC;
TIMEZONEADJUSTMENT=-240;
APPINFO='Interactive SQL connection'
```

AutoStart 连接参数 [ASTART]

控制在找不到连接时是否启动本地数据库服务器。

用法

任何位置

值

YES、NO

缺省值

YES

注释

缺省情况下，如果在尝试建立连接时找不到服务器，并且指定了数据库文件、数据库名或 START 连接参数，则启动同一台计算机上的数据库服务器。您可以通过将连接字符串中的 AutoStart (ASTART) 连接参数设置为 NO 来关闭此行为。如果 CommLinks [LINKS] 参数包含 TCPIP，数据库服务器将不会自动启动。

为了提高自动启动数据库的查询性能，请尽快启动数据库，即使用户不会立即与数据库建立连接。这样，可在对数据库执行查询前，使高速缓存预热。请参见“[使用高速缓存预热](#)”一节《[SQL Anywhere 服务器 - SQL 的用法](#)》。

另请参见

- [“连接参数”一节第 80 页](#)
- [“解决连接参数冲突”一节第 81 页](#)
- [“查找数据库服务器”一节第 126 页](#)
- [“CommLinks 连接参数 \[LINKS\]”一节第 254 页](#)
- [“Elevate 连接参数”一节第 264 页](#)

AutoStop 连接参数 [ASTOP]

控制在没有打开的非 HTTP 连接时是否停止数据库。

用法

嵌入式数据库

值

YES、NO

缺省值

YES

注释

缺省情况下，从连接字符串启动的任何数据库服务器在不再有非 HTTP 连接时都会停止。同样地，从连接字符串装载的任何数据库在不再有非 HTTP 连接时都会卸载。该行为等同于 AutoStop=YES。

如果设置 AutoStop=NO，在该连接中启动的任何数据库在不再有非 HTTP 连接时，仍将保持运行。因此，该数据库服务器也同样保持运行。

如果与数据库的唯一连接是 HTTP 连接而且该数据库配置为自动停止，则当该 HTTP 连接断开时，数据库不会自动停止。同样，如果配置为自动停止的数据库具有 HTTP 连接和命令序列或 TDS 连接，则当最后一个命令序列或 TDS 连接断开时，该数据库就会自动停止，并且所有 HTTP 连接都将断开。请参见“[-ga 服务器选项](#)”一节第 177 页和“[AutoStop 连接参数 \[ASTOP\]](#)”一节第 251 页。

只有在连接当前未运行的数据库时，才使用 AutoStop (ASTOP) 连接参数。如果数据库已启动，则忽略该参数。

在 .NET 应用程序中，使用 AutoStop 连接参数时应小心。就该应用程序而言，关闭连接会将该参数关闭，但启用连接池时活动连接会保持打开状态。因此，尽管您希望服务器关闭，但它不会关闭。

另请参见

- [“连接参数”一节第 80 页](#)
- [“解决连接参数冲突”一节第 81 页](#)
- [“连接池”一节《SQL Anywhere 服务器 - 编程》](#)
- [“启动和停止数据库”一节第 54 页](#)
- [“START DATABASE 语句”一节《SQL Anywhere 服务器 - SQL 参考》](#)

CharSet 连接参数 [CS]

指定要在此连接上使用的字符集。

用法

任何位置

值

字符串

缺省值

本地字符集。

有关如何确定本地字符集的详细信息，请参见“[确定区域设置信息](#)”一节第 399 页。

注释

如果为 CharSet 提供一个值，则指定的字符集将用于当前的连接。设置 CharSet=none 将禁用为连接进行字符集转换。

卸载数据时，可使用 CharSet 连接参数指定字符集。有关有效字符集值的详细信息，请参见“[建议的字符集和归类](#)”一节第 408 页。

为避免字符集转换出现损失，在使用 Unicode 客户端 API 时，建议不要设置 CHARSET 连接参数。Unicode 客户端 API 包括 ADO.NET、OLE DB 和 iAnywhere JDBC 驱动程序。当使用 wide (Unicode) 函数时，ODBC 同样也是 Unicode 客户端 API。

另请参见

- [“连接参数”一节第 80 页](#)
- [“解决连接参数冲突”一节第 81 页](#)
- [“SACHARSET 环境变量”一节第 352 页](#)
- [“了解区域设置字符集”一节第 391 页](#)

CommBufferSize 连接参数 [CBSIZE]

设置通信包的最大大小（以字节为单位）。

用法

任何位置

值

Integer [**k**]

缺省值

如果未设置 CommBufferSize 值，则 CommBufferSize 由服务器上的设置（在除 Windows Mobile 外的所有操作系统上缺省值为 7300 字节）控制。在 Windows Mobile 上，缺省值为 1460 字节。

注释

CommBufferSize (CBSIZE) 连接参数用于指定通信包大小（以字节为单位）。使用 **k** 指定以千字节为单位。CommBufferSize 的最小值是 500 字节，最大值是 16000 字节。

协议栈决定了网络上包的最大大小。如果设置的 CommBufferSize 值大于网络所允许的值，则通信包由网络软件拆分。缺省值为标准以太网 TCP/IP 最大数据包大小（1460 字节）的倍数。

较大的数据包大小可以提高多行读取和较大行读取的性能，但同时会增加客户端和服务器的内存使用。

如果未在客户端指定 CommBufferSize，则连接使用服务器的缓冲区大小。如果在客户端指定 CommBufferSize，则连接使用该 CommBufferSize 值。

使用 **-p** 数据库服务器选项设置 CommBufferSize 时，会让所有未指定自己的 CommBufferSize 值的客户端都使用 **-p** 数据库服务器选项指定的大小。

另请参见

- [“连接参数”一节第 80 页](#)
- [“解决连接参数冲突”一节第 81 页](#)
- [“调整 TCP/IP 性能”一节第 136 页](#)
- [“-p 服务器选项”一节第 200 页](#)

示例

将缓冲区的大小设置为 1460 字节：

```
...  
CommBufferSize=1460  
...
```

您也可以可以在 [SQL Anywhere 11 的 ODBC 配置] 窗口中 [网络] 选项卡上的 [缓冲区大小] 文本框中输入参数值来设置此参数。

CommLinks 连接参数 [LINKS]

指定客户端网络协议选项。

用法

任何地方。CommLinks (LINKS) 连接参数是个人服务器连接的可选参数，而是网络服务器连接的必需参数。

值

字符串

缺省值

只使用共享内存通信协议来连接。

注释

如果未指定 CommLinks (LINKS) 连接参数，则客户端只在当前计算机上并只使用共享内存连接来搜索服务器。这是缺省行为，该行为等同于 CommLinks=ShMem。共享内存协议是在同一台计算机上运行的客户端和服务器之间最快的通信链接，通常适用于与个人数据库服务器连接的应用程序。

有关保护 Unix 上共享内存连接的信息，请参见“安全提示”一节第 1005 页。

如果指定 CommLinks=ALL，则客户端会使用所有可用通信协议来搜索服务器。由于指定 CommLinks=ALL 时可能会影响系统性能，因此在不知道使用哪个协议时才使用此设置。

如果在 CommLinks (LINKS) 连接参数中指定一个或多个协议，则客户端将按指定的顺序使用指定的通信协议搜索网络数据库服务器。请注意，如果已指定共享内存，则首先尝试使用共享内存进行连接，然后按照指定的顺序尝试其余的通信协议。如果连接未能使用某个指定的协议进行连接，则会出现连接错误并且连接尝试会中止，即使列表中仍有可尝试的协议。

CommLinks (LINKS) 连接参数的值不区分大小写，包括：

- **SharedMemory (ShMem)** 启动同机通信的共享内存协议。这是缺省设置。如果在协议列表中包含该协议，则无论协议显示的顺序如何，客户端都会首先尝试使用共享内存。
- **ALL** 首先尝试使用共享内存协议进行连接，然后使用所有其余的可用通信协议进行连接。如果无法确定使用哪个（些）协议，则使用此设置。
- **TCPIP (TCP)** 启动 TCP/IP 通信协议。所有操作系统都支持 TCP/IP。如果 CommLinks [LINKS] 参数包含 TCPIP，个人数据库服务器将不会自动启动。

其中的每个值都可以提供附加网络协议选项。

请参见“网络协议选项”一节第 284 页。

由于以下原因，您可能需要使用特定协议，而不是 [ALL] 协议：

- 如果客户端仅使用必需的网络协议，则网络库的启动会稍微快一些。
- 与数据库的连接速度可能会快些。
- 如果您要通过提供其它网络协议选项来调整特定协议的广播行为，则必须显式指定该协议。

CommLinks (LINKS) 连接参数与数据库服务器 -x 选项相对应。

另请参见

- “网络协议选项” 一节第 284 页
- “客户端/服务器通信” 第 133 页
- “-x 服务器选项” 一节第 221 页
- “连接参数” 一节第 80 页
- “解决连接参数冲突” 一节第 81 页
- “高速缓存服务器名以提高连接速度” 一节第 129 页
- CommLinks 属性：“连接属性” 一节第 566 页

示例

以下连接字符串片段只启动 TCP/IP 协议：

```
CommLinks=tcPIP
```

以下连接字符串片段启动共享内存协议并通过共享内存搜索数据库服务器。如果搜索失败，它将启动 TCP/IP 协议并在本地网络上搜索该服务器。

```
CommLinks=tcPIP,shmem
```

以下连接字符串片段启动共享内存协议并通过共享内存搜索服务器。如果搜索失败，将启动 TCP 协议并在本地网络和主机 kangaroo 上搜索服务器。请注意，如果通过共享内存找到了该服务器，则不启动 TCP 链接。

```
CommLinks=shmem,tcPIP(HOST=kangaroo)
```

Compress 连接参数 [COMP]

为连接打开或关闭压缩功能。在某些情况下，压缩连接可以提高性能。

用法

除 TDS 连接以外的任何位置。TDS 连接（包括 jConnect）不支持 SQL Anywhere 通信压缩。

值

YES、NO

如果客户端设置和服务器设置不同，则应用客户端设置。

缺省值

NO

如果未设置 `Compress` 连接参数的值，则压缩状态由服务器上的设置控制，其缺省状态为不压缩。

注释

在 SQL Anywhere 客户端和服务器之间发送的包可通过使用 `Compress (COMP)` 连接参数进行压缩。含有高度可压缩数据的大数据量传输通常可获得最佳的压缩率。

指定 `YES` 或 `NO` 可为连接打开或关闭通信压缩功能。它们不区分大小写。

在生产环境中使用通信压缩之前，建议在该网络上使用特定的应用程序进行性能分析。

要针对服务器上的所有远程连接启用压缩，请使用 `-pc` 服务器选项。

请注意，同一台计算机上通过任何通信链接建立的连接不会启用压缩，即使使用 `-pc` 选项或 `COMPRESS=YES` 参数也是如此。

另请参见

- [“连接参数”一节第 80 页](#)
- [“解决连接参数冲突”一节第 81 页](#)
- [“-pc 服务器选项”一节第 200 页](#)
- [“调整通信压缩设置以改善性能”一节第 141 页](#)
- [“使用压缩功能”一节 《SQL Anywhere 服务器 - SQL 的用法》](#)

示例

以下连接字符串片段打开包压缩：

```
Compress=YES
```

以下连接字符串片段关闭包压缩功能：

```
Compress=NO
```

CompressionThreshold 连接参数 [COMPPTH]

增大或减小压缩包的大小限制。通过更改压缩阈值，您可以只在压缩会提高数据包的传输速度时才对数据包进行压缩，从而提高压缩连接的性能。

用法

除 TDS 以外的任何位置。只适用于压缩连接。

值

Integer [**k**]

如果客户端和服务器指定了不同的压缩阈值设置，则使用客户端设置。

缺省值

120

如果未设置 `CompressionThreshold` 值，则压缩阈值由服务器上的设置（缺省值为 120 字节）控制。

注释

如果启用了压缩功能，则单个数据包可能会压缩也可能会不压缩，具体情况视其大小而定。例如，即使启用了通信压缩，SQL Anywhere 也不压缩小于压缩阈值的数据包。此外，小数据包（小于 100 字节）通常不压缩。由于压缩数据包时需要 CPU 时间，因此尝试压缩小数据包实际上可能会降低性能。

该值表示要压缩数据包的最小大小（以字节为单位）。使用 **k** 指定以千字节为单位。所支持的最小值为 1 个字节，所支持的最大值为 32767 字节。建议不要使用小于 80 字节的值。

一般来说，降低压缩阈值可能会改善速度非常慢的网络的性能，而提高压缩阈值可能会减少 CPU 负载，从而使性能得到改善。但是，由于降低压缩阈值在客户端和服务器的上均会增加 CPU 使用率，因此应进行性能分析以确定更改压缩阈值是否有好处。

另请参见

- [“-pt 服务器选项”一节第 201 页](#)
- [“调整通信压缩设置以改善性能”一节第 141 页](#)
- [“连接参数”一节第 80 页](#)
- [“解决连接参数冲突”一节第 81 页](#)

示例

连接，压缩阈值为 100 个字节。

```
CompressionThreshold=100
```

ConnectionName 连接参数 [CON]

对连接进行命名，便于在多连接应用程序中切换到它。

用法

任何位置

值

字符串

缺省值

无连接名称。

注释

可选参数，为所建立的特定连接指定名称。除非您要建立多个连接并在它们之间进行切换，否则不要指定此参数。

连接名称与数据源名称不同。

另请参见

- “连接参数”一节第 80 页
- “解决连接参数冲突”一节第 81 页
- “SET CONNECTION 语句 [Interactive SQL] [ESQL]”一节 《SQL Anywhere 服务器 - SQL 参考》

示例

连接，指定连接名为 first-con:

```
CON=first-con
```

DatabaseFile 连接参数 [DBF]

指示在启动未运行的数据库时所要装载和连接的数据库文件。

如果要连接到已在运行的数据库，可使用 DatabaseName (DBN) 参数。

用法

嵌入式数据库

值

字符串

缺省值

无缺省设置。

注释

DatabaseFile (DBF) 连接参数用于装载并连接到未在数据库服务器上运行的指定数据库文件。

- 如果要连接到的数据库未运行，请使用 DatabaseFile (DBF) 连接参数启动数据库。
- 如果文件名中不包含扩展名，SQL Anywhere 就会查找扩展名为 *.db* 的文件。
- 文件的路径与数据库服务器的工作目录相关。如果您从命令提示符启动服务器，则工作目录是您在输入该命令时所在的目录。如果您从图标或快捷方式启动服务器，则它是该图标或快捷方式所指定的工作目录。建议提供完整路径和文件名。
- 如果同时指定数据库文件和数据库名，则首先尝试以所指定的名称（忽略数据库文件）连接正在运行的数据库，如果失败，则尝试同时使用数据库文件和数据库名自动启动数据库。如果 CommLinks [LINKS] 参数包含 TCPIP，数据库服务器将不会自动启动。

您也可以使用 UNC 文件名。

有关使用 UNC 文件的详细信息，请参见“[SQL Anywhere 数据库服务器](#)”一节第 148 页。

当尝试自动启动一个未运行的数据库文件时，建议所部署的应用程序使用 ServerName (ENG) 参数来指定数据库服务器名称。否则，应用程序可能会意外连接到其它数据库服务器上。例如，数据库服务器可能连接到属于某个嵌入式应用程序并且已在运行的不同版本 SQL Anywhere 服务器上。

小心

数据库文件必须位于数据库服务器所在的计算机上。启动位于网络驱动器上的数据库文件可能导致文件损坏。

另请参见

- “-gd 服务器选项” 一节第 179 页
- “CommLinks 连接参数 [LINKS]” 一节第 254 页
- “DatabaseName 连接参数 [DBN]” 一节第 260 页
- “连接参数” 一节第 80 页
- “解决连接参数冲突” 一节第 81 页
- “连接到嵌入式数据库” 一节第 118 页

示例

以下示例中的 DatabaseFile (DBF) 连接参数会装载并连接到示例数据库 *demo.db*:

```
DBF=samples-dir\demo.db
```

有关 *samples-dir* 的信息，请参见“[示例目录](#)”一节第 368 页。

以下两个示例假定您已经启动名为 *cities.db* 的数据库文件，并将数据库重命名为 *Kitchener*，如下所示：

```
dbeng11 cities.db -n Kitchener
```

要成功地启动和连接数据库并将其命名为 *Kitchener*：

```
DBN=Kitchener;DBF=cities.db
```

如果指定 `DBF=cities.db`，则无法连接正在运行的名为 *Kitchener* 的数据库。

DatabaseKey 连接参数 [DBKEY]

以连接请求启动加密数据库。

用法

任何位置

值

字符串

缺省值

无

注释

以连接请求启动加密数据库时，必须指定此参数。如果要连接到已运行的加密数据库，则无需指定此参数。

加密密钥是一个包括混合大小写、数字、字母和特殊字符的字符串。数据库密钥中不能包含前导空格、尾随空格或分号。

若要保证客户端应用程序与数据库服务器之间通信包的安全，可使用 `-ec` 服务器选项和传送层安全。请参见“[传送层安全](#)”第 1031 页。

另请参见

- “[配置客户端应用程序以使用传送层安全](#)”一节第 1043 页
- “[-ec 服务器选项](#)”一节第 171 页
- “[-ek 数据库选项](#)”一节第 238 页
- “[-ep 服务器选项](#)”一节第 173 页
- “[-es 服务器选项](#)”一节第 174 页
- “[连接参数](#)”一节第 80 页
- “[解决连接参数冲突](#)”一节第 81 页
- “[Encryption 连接参数 \[ENC\]](#)”一节第 265 页

示例

以下片段说明 DatabaseKey (DBKEY) 连接参数的用法：

```
"UID=DBA;PWD=sql;ENG=myeng;DBKEY=V3moj3952B;DBF=samples-dir\demo.db"
```

DatabaseName 连接参数 [DBN]

标识在连接已运行的数据库时需要与之建立连接的已装载数据库。

如果要连接到未运行的数据库，可使用 DatabaseFile (DBF) 参数。

用法

运行本地数据库或网络服务器

值

字符串

缺省值

无缺省设置。

注释

任何时候在服务器上启动数据库时，都会为它指派一个数据库名，其方法是管理员使用 `-n` 选项，或者是服务器使用去掉了扩展名和路径的基本文件名。

只能使用数据库名 `utility_db` 连接 SQL Anywhere 实用程序数据库。请参见“[使用实用程序数据库](#)”一节第 28 页。

注意

命名数据库时建议使用 DatabaseName (DBN) 连接参数，而不要使用 DatabaseSwitches (DBS) 连接参数的 `-n` 选项。

如果要连接到的数据库已在运行，则应该指定数据库名而不是数据库文件。

只有运行的数据库的名称与 `DatabaseName (DBN)` 参数中指定的名称匹配时，连接才会发生。

注意

如果同时指定数据库文件和数据库名，则使用指定名称（忽略数据库文件）尝试连接正在运行的数据库，如果失败，则同时使用数据库文件和数据库名尝试自动启动数据库。

另请参见

- “连接参数”一节第 80 页
- “解决连接参数冲突”一节第 81 页
- “DatabaseName 协议选项 [DBN]”一节第 291 页

示例

要启动名为 `cities.db` 的数据库文件并将数据库重命名为 `Kitchener`，可以使用以下命令：

```
dbeng11 cities.db -n Kitchener
```

如果已运行上述命令，则可成功地连接到正在运行的名为 `Kitchener` 的数据库，如下所示：

```
DBN=Kitchener
```

也可以使用以下命令成功地连接到正在运行的名为 `Kitchener` 的数据库：

```
DBN=Kitchener;DBF=cities.db
```

但是，指定以下参数将无法连接到名为 `Kitchener` 的数据库：

```
DBF=cities.db
```

DatabaseSwitches 连接参数 [DBS]

在启动数据库时提供数据库特定的选项。

用法

在未装载数据库时连接到服务器。如果数据库服务器未运行，此连接参数将使用指定的数据库和选项自动启动服务器。

值

字符串

缺省值

无选项。

注释

只有在您要连接到当前未运行的数据库时，才应当设置 `DatabaseSwitches`。当服务器启动由 `DatabaseFile` 指定的数据库时，服务器会使用提供的 `DatabaseSwitches` 参数确定该数据库的启动选项。

使用此参数只能提供 *数据库* 选项。服务器选项则必须使用 `StartLine` 连接参数提供。

请参见“[数据库选项](#)”一节第 234 页。

注意

命名数据库时建议使用 `DatabaseName (DBN)` 连接参数，而不要使用 `DatabaseSwitches (DBS)` 连接参数的 `-n` 选项。

另请参见

- “[SQL Anywhere 数据库服务器](#)”一节第 148 页
- “[StartLine 连接参数 \[START\]](#)”一节第 281 页
- “[连接参数](#)”一节第 80 页
- “[解决连接参数冲突](#)”一节第 81 页
- “[DatabaseName 连接参数 \[DBN\]](#)”一节第 260 页

示例

在命令提示符下输入到一行中的以下命令会连接缺省数据库服务器，装载数据库文件 `demo.db` (`DatabaseFile (DBF)` 连接参数)，将其命名为 `my-db` (`DatabaseName (DBN)` 连接参数) 并在只读模式 (`-r` 选项) 下将其启动。

```
dbisql -c "UID=DBA;PWD=sql;DBF=samples-dir\demo.db;DBN=my-db;DBS=-r"
```

有关 `samples-dir` 的信息，请参见“[示例目录](#)”一节第 368 页。

DataSourceName 连接参数 [DSN]

通知 ODBC 驱动程序管理器或嵌入式 SQL 库在注册表或系统信息文件（缺省情况下名为 `.odbc.ini`）的何处查找 ODBC 数据源信息。

用法

任何位置

值

字符串

缺省值

没有缺省数据源名称。

注释

对于 ODBC 应用程序来说，通常的做法是只向 ODBC 发送数据源名称。ODBC 驱动程序管理器和 ODBC 驱动程序将查找包含其余连接参数的数据源。

在 SQL Anywhere 中，嵌入式 SQL 应用程序也可以使用 ODBC 数据源存储连接参数。

另请参见

- [“FileDataSourceName 连接参数 \[FILEDSN\]”](#) 一节第 267 页
- [“连接参数”](#) 一节第 80 页
- [“解决连接参数冲突”](#) 一节第 81 页
- [“在 Unix 上使用 ODBC 数据源”](#) 一节第 95 页
- [“创建 ODBC 数据源”](#) 一节第 90 页

示例

以下参数使用一个数据源名：

```
DSN=My Database
```

DisableMultiRowFetch 连接参数 [DMRF]

关闭跨网络的多行读取。

用法

任何位置

值

YES、NO

缺省值

NO

注释

缺省情况下，在数据库服务器收到一个简单读取请求时，应用程序会请求更多的行。您可以通过将此参数设置为 YES 来禁用此行为。

请参见 [“在过程和触发器中使用游标”](#) 一节 《SQL Anywhere 服务器 - SQL 的用法》。

将 DisableMultiRowFetch (DMRF) 连接参数设置为 YES 等同于将预取数据库选项设置为 [Off]。

请参见 [“预取行”](#) 一节 《SQL Anywhere 服务器 - 编程》。

另请参见

- [“连接参数”](#) 一节第 80 页
- [“解决连接参数冲突”](#) 一节第 81 页
- [“prefetch 选项 \[数据库\]”](#) 一节第 534 页

示例

以下连接字符串片段禁止预读：

```
DMRF=YES
```

Elevate 连接参数

在 Windows Vista 上自动提升自动启动的数据库服务器执行文件。

用法

仅限 Windows Vista

值

YES、NO

缺省值

NO

注释

可以在连接字符串中指定 ELEVATE=YES，以便提升自动启动的数据库服务器执行文件。这样，未提升的客户端进程就可以自动启动提升的服务器，这是 Windows Vista 中必需的操作，因为未提升的服务器不能使用 AWE 内存。如果数据库服务器未自动启动，则忽略该参数。启动数据库服务器命令时必须指定 -cw 选项，才能使用 AWE 高速缓存。

另请参见

- [“-cm 服务器选项”一节第 163 页](#)
- [“-cw 服务器选项”一节第 167 页](#)

示例

以下连接字符串片段会使自动启动的数据库服务器在 Windows Vista 中获得提升，以便它们可以使用 AWE 高速缓存：

```
"Elevate=YES;START=dbeng11 -cw"
```

EncryptedPassword 连接参数 [ENP]

提供口令，以加密形式存储在数据源中。

用法

任何位置

值

字符串

缺省值

无

注释

小心

数据源以文件形式存储在磁盘上或存储在注册表中。以明文形式存储口令，或者使用加密口令都存在很大的安全风险。数据库中存储有任何敏感数据时，建议不要这么做。当您将口令输入到数据源中时，可以用加密形式存储，以使安全性比明文形式存储时高一些。

在 Unix 系统中，此信息存储在系统信息文件（缺省情况下名为 *.odbc.ini*）中。

有关如何查找系统信息文件的详细信息，请参见“[在 Unix 上使用 ODBC 数据源](#)”一节第 95 页。

如果 Password (PWD) 连接参数和 EncryptedPassword (ENP) 连接参数都指定了，则 Password (PWD) 优先。

另请参见

- “[连接参数](#)”一节第 80 页
- “[解决连接参数冲突](#)”一节第 81 页
- “[Password 连接参数 \[PWD\]](#)”一节第 276 页

Encryption 连接参数 [ENC]

使用传送层安全性或简单加密方式加密在客户端应用程序和服务器之间发送的数据包。

用法

TLS: 仅受 TCP/IP 支持

NONE 或 SIMPLE: 所有情况下都支持

值

```
Encryption= { NONE
| SIMPLE
| TLS( TLS_TYPE=cipher;
  [ FIPS={ Y | N }; ]
  TRUSTED_CERTIFICATES=public-certificate;
  [ CERTIFICATE_COMPANY=organization; ]
  [ CERTIFICATE_NAME=common-name; ]
  [ CERTIFICATE_UNIT=organization-unit ] )
```

缺省值

NONE

注释

如果要使用传送层安全性和简单加密来保护客户端应用程序与数据库服务器之间通信的安全，则可使用此参数。请参见“[传送层安全](#)”第 1031 页。

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》。

Encryption (ENC) 连接参数接受以下参数:

- **NONE** 接受未加密的通信包。
- **SIMPLE** 接受用所有平台上和早期版本的 SQL Anywhere 上支持的简单加密进行加密的通信包。简单加密不提供服务器验证、高度椭圆曲线、RSA 加密或其它传送层安全性功能。
如果数据库服务器接受简单加密，但不接受无加密，则所有使用无加密的非 TDS 连接尝试会自动使用简单加密。
使用 `-ec SIMPLE` 启动数据库服务器将通知数据库服务器仅接受使用简单加密的连接。TLS 连接 (ECC、RSA、RSA FIPS) 将失败，并且无加密要求的连接会使用简单加密。
使用 `-ec SIMPLE, TLS (TLS_TYPE=ECC; ...)` 启动数据库服务器将通知数据库服务器仅接受使用 ECC TLS 加密或简单加密的连接。RSA 和 RSA FIPS 连接都失败并且无加密要求的连接会使用简单加密。
- **cipher** 对于 RSA 和 ECC 加密而言，可分别为 **RSA** 或 **ECC**。对于 FIPS 认可的 RSA 加密，请指定 **TLS_TYPE=RSA;FIPS=Y**。RSA FIPS 使用单独的认可库，但是与使用 SQL Anywhere 9.0.2 或更高版本指定 RSA 的服务器兼容。

如果编码器与用于创建证书的加密 (RSA 或 ECC) 不匹配，则连接会失败。

客户端可以使用以下参数来验证服务器公共证书中的字段值。

- `trusted_certificates`
- `certificate_company`
- `certificate_unit`
- `certificate_name`

有关针对服务器验证进行证书字段验证的详细信息，请参见“校证书字段”一节第 1044 页。

有关使用数字证书的详细信息，请参见“创建数字证书”一节第 1036 页。

可以使用 `CONNECTION_PROPERTY` 系统函数来检索当前连接的加密设置:

```
SELECT CONNECTION_PROPERTY ( 'Encryption' );
```

该函数返回以下五个值之一: `None`、`Simple`、`ecc_tls`、`rsa_tls` 或 `rsa_tls_fips`，具体取决于连接所使用的加密类型。

请参见“`CONNECTION_PROPERTY` 函数 [System]”一节《SQL Anywhere 服务器 - SQL 参考》。

另请参见

- “配置客户端应用程序以使用传送层安全”一节第 1043 页
- “-ec 服务器选项”一节第 171 页
- “-ek 数据库选项”一节第 238 页
- “-ep 服务器选项”一节第 173 页
- “-es 服务器选项”一节第 174 页
- “连接参数”一节第 80 页
- “解决连接参数冲突”一节第 81 页
- “DatabaseKey 连接参数 [DBKEY]”一节第 259 页
- “certificate_company 协议选项”一节第 287 页
- “certificate_name 协议选项”一节第 288 页
- “certificate_unit 协议选项”一节第 289 页
- “trusted_certificates 协议选项”一节第 307 页

示例

以下连接字符串片段使用传送层安全性和椭圆曲线加密，通过 TCP/IP 链接连接到名为 demo 的数据库服务器：

```
"ENG=demo;LINKS=tcip;ENCRYPTION=tls(tls_type=ecc;trusted_certificates=eccroot.crt)"
```

以下连接字符串片段使用传送层安全性和 RSA 加密，通过 TCP/IP 链接连接到名为 demo 的数据库服务器：

```
"ENG=demo;LINKS=tcip;ENCRYPTION=tls(tls_type=rsa;fips=n;trusted_certificates=rsaroot.crt)"
```

以下连接字符串片段使用简单加密，通过 TCP/IP 链接连接到名为 demo 的数据库服务器：

```
"ENG=demo;LINKS=tcip;ENCRYPTION=simple"
```

EngineName 连接参数 [ENG]

这是 ServerName (ENG) 连接参数的同义词。请参见“ServerName 连接参数 [ENG]”一节第 280 页。

FileDataSourceName 连接参数 [FILEDSN]

通知客户端库有一个 ODBC 文件数据源保存有关要连接到的数据库的信息。

用法

任何位置

值

字符串

缺省值

无缺省名。

注释

文件数据源中保存的信息与注册表中的 ODBC 数据源所保存的信息相同。文件数据源可方便地发到最终用户，这样就不必在每台计算机上重建连接信息。

ODBC 和嵌入式 SQL 应用程序均可使用文件数据源。

另请参见

- [“DataSourceName 连接参数 \[DSN\]”一节第 262 页](#)
- [“连接参数”一节第 80 页](#)
- [“解决连接参数冲突”一节第 81 页](#)
- [“在 Windows 上使用文件数据源”一节第 94 页](#)

ForceStart 连接参数 [FORCE]

启动数据库服务器，但不尝试连接数据库服务器。

用法

只用于 db_start_engine 函数。

值

YES、NO

缺省值

NO

注释

通过设置 ForceStart=YES，db_start_engine 函数启动一个服务器但并不试图连接到一个服务器，即使已有一个服务器在运行也是如此。

另请参见

- [“db_start_engine 函数”一节《SQL Anywhere 服务器 - 编程》](#)
- [“连接参数”一节第 80 页](#)
- [“解决连接参数冲突”一节第 81 页](#)

Idle 连接参数

指定连接的空闲超时时间。

用法

除 TDS 连接和共享内存连接以外的任何位置。共享内存连接和 TDS 连接（包括 jConnect）会忽略 SQL Anywhere 的空闲 (IDLE) 连接参数。

值

整数

缺省值

无

注释

Idle (IDLE) 连接参数只适用于当前连接。您可以为同一台服务器上的多个连接设置不同的超时值。

如果未设置连接空闲超时值，则空闲超时值由服务器上的设置（缺省值为 240 分钟）控制。如果超时值之间有冲突，则无论是否指定服务器超时值，连接超时值都会取代任何服务器超时值。

IDLE 连接参数的最小超时值为 1 分钟，所支持的最大超时值为 32767 分钟。如果将超时值指定为 0，则对连接的空闲超时值检查将被关闭。

另请参见

- [“-ti 服务器选项”一节第 212 页](#)
- [“连接参数”一节第 80 页](#)
- [“解决连接参数冲突”一节第 81 页](#)
- [“调整超时值”一节第 145 页](#)

示例

以下连接字符串片段将此连接的超时值设置为 10 分钟：

```
"ENG=myeng;LINKS=tcPIP;IDLE=10"
```

Integrated 连接参数 [INT]

指定是否可尝试集成登录。

用法

任何位置

值

YES、NO

缺省值

NO

注释

Integrated (INT) 连接参数具有以下设置：

- **YES** 尝试进行集成登录。如果连接尝试失败且 `login_mode` 选项设置为 `[Standard, Integrated]`，则尝试进行标准登录。
- **NO** 这是缺省设置。不尝试集成登录。

对于要使用集成登录的客户端应用程序，服务器必须在将 `login_mode` 数据库选项设置为包括 `[Integrated]` 在内的值的情况下运行。

另请参见

- [“login_mode 选项 \[数据库\]”一节第 512 页](#)
- [“连接参数”一节第 80 页](#)
- [“解决连接参数冲突”一节第 81 页](#)
- [“使用集成登录”一节第 99 页](#)

示例

以下数据源片段使用集成登录：

```
INT=YES
```

Kerberos 连接参数 [KRB]

指定在连接数据库服务器时是否可以使用 Kerberos 验证。

用法

除 Windows Mobile 之外的所有平台。

值

YES、NO、SSPI 或 *GSS-API-library-file*

缺省值

NO

注释

Kerberos [KRB] 连接参数具有以下设置：

- **YES** 尝试进行 Kerberos 验证的登录。
- **NO** 不尝试进行 Kerberos 验证的登录。这是缺省设置。
- **SSPI** 尝试进行 Kerberos 验证的登录，并且使用内置的 Windows SSPI 接口代替 GSS-API 库。SSPI 仅可用于 Windows 平台，并且不能用于除了域控制器 Active Directory KDC 之外的密钥分发中心 (KDC)。如果您的 Windows 客户端计算机已经登录到某个 Windows 域，则可使用 SSPI 而无需安装或配置 Kerberos 客户端。
- **GSS-API 库文件** 尝试进行 Kerberos 验证登录，此字符串指定 Kerberos GSS-API 库的文件名（或者是 Unix 上的共享对象）。仅在 Kerberos 客户端对 Kerberos GSS-API 库使用与缺省文件名不同的文件名时或在计算机上安装了多个 GSS-API 库时，才需要此设置。

使用 Kerberos 验证登录时，会忽略 UserID 和 Password 连接参数。

要使用 Kerberos 验证，必须已经安装并配置 Kerberos 客户端（对于 SSPI 不需进行任何准备），用户必须已经登录到 Kerberos（具有一个有效的票据授予票据），而且必须已经启用数据库服务器并配置 Kerberos 验证登录。

另请参见

- [“-kl 服务器选项”一节第 190 页](#)
- [“-kr 服务器选项”一节第 191 页](#)
- [“-krb 服务器选项”一节第 192 页](#)
- [“Kerberos 验证”一节第 107 页](#)
- [“GRANT 语句”一节《SQL Anywhere 服务器 - SQL 参考》](#)
- [“在 Windows 上使用 SSPI 进行 Kerberos 登录”一节第 112 页](#)

示例

```
Kerberos=YES  
Kerberos=SSPI  
Kerberos=c:\Program Files\MIT\Kerberos\bin\gssapi32.dll
```

Language 连接参数 [LANG]

指定连接的语言。

用法

任何位置

值

代表语言的双字母组合。例如，指定 LANG=DE 可将缺省语言设置为德语。

缺省值

由 SALANG 环境变量、dblang 实用程序或安装程序指定的语言（按顺序）。

注释

此连接参数用于确定连接的语言。来自服务器的任何错误或警告都以指定的语言（假定该服务器支持该语言）进行传送。

如果未指定任何语言，则使用缺省语言。缺省语言是依次由 SALANG 环境变量、dblang 实用程序或安装程序指定的语言。

有关语言代码的详细信息，请参见 [“了解区域设置语言”一节第 389 页](#)。

此连接参数仅影响该连接。从 SQL Anywhere 的工具和实用程序返回的消息会以缺省语言显示，而从服务器返回的消息则以连接的语言显示。

另请参见

- [“连接参数”一节第 80 页](#)
- [“解决连接参数冲突”一节第 81 页](#)

LazyClose 连接参数 [LCLOSE]

控制游标请求是一直排队等到下一个请求，还是立即执行。对关闭游标请求进行排队可节省往返时间，提高性能。

用法

任何位置

值

YES、NO、AUTO

缺省值

AUTO

注释

- **YES** 始终对游标关闭请求进行排队，这可以节省往返时间，但在客户端关闭该游标后会占用锁和其它资源。当下一个请求发送到同一连接上的数据库服务器时，将执行游标关闭。当 `CLOSE cursor-name` 数据库请求处于排队状态时，任何隔离级别为 1 的游标稳定性锁仍旧适用于该游标。
- **NO** 立即关闭游标。
- **AUTO** 对游标关闭请求进行排队，从而节省往返时间，仅在不改变锁定时间或占用大量服务器资源的时间时才这么做。如果游标使用隔离级别为 1 的游标稳定性锁，或者会消耗大量在关闭该游标后才释放的服务器资源，则立即关闭该游标。需要工作表的查询就是一个会消耗大量服务器资源的游标示例。

如果此连接参数设置为 YES 或 AUTO，则在出现下一个数据库请求时才会关闭游标。

如果您的网络等待时间较长，或者应用程序发送许多游标打开和关闭请求，则启用此选项可以提高性能。

另请参见

- “连接参数”一节第 80 页
- “解决连接参数冲突”一节第 81 页
- “减少客户端和服务器之间的请求”一节 《SQL Anywhere 服务器 - SQL 的用法》

LivenessTimeout 连接参数 [LTO]

在连接出现异常时控制连接的关闭。

用法

仅限网络服务器。

除了非线程 Unix 应用程序外的所有平台。

值

整数（以秒为单位）

缺省值

无

如果未设置 LivenessTimeout 值，则 LivenessTimeout 由服务器上的设置（缺省值为 120 秒）控制。

注释

活动数据包通过客户端/服务器 TCP/IP 通信协议定期发送，用以确认连接的完整性。如果客户端的运行时间达到了 LivenessTimeout 周期但没有检测到活动请求或响应数据包，通信将终止。

如果连接在 LivenessTimeout 值的三分之一到三分之二这么长的一段时间内未发送任何数据包，则将发送活动数据包。

当与服务器的连接超过 200 个时，服务器会根据指定的 LivenessTimeout 值自动计算一个更大的 LivenessTimeout 值。这使服务器可以更有效地处理大量连接。

您也可以在 [SQL Anywhere 11 的 ODBC 配置] 窗口中 [网络] 选项卡上的 [活动超时] 文本框中输入参数值来设置此参数。

LivenessTimeout 连接参数的最小值为 30 秒，最大值为 32767 秒。如果将超时值指定为 0，则对连接的活动超时值检查将被关闭。任何小于最小值的非零值将被重置为最小值。例如，包含 "LivenessTimeout=5" 的连接字符串使用 "LivenessTimeout=30" 代替。

另请参见

- [“连接参数”一节第 80 页](#)
- [“解决连接参数冲突”一节第 81 页](#)
- [“-tl 服务器选项”一节第 212 页](#)

示例

以下连接字符串片段将 LivenessTimeout 值设置为 10 分钟：

```
LTO=600
```

Logfile 连接参数 [LOG]

将客户端错误消息和调试消息发送到文件中。

用法

任何位置

值

字符串

缺省值

无日志文件。

注释

如果要将客户端错误消息和调试消息保存到文件中，请使用 LogFile (LOG) 连接参数。

如果文件名中不包括路径，则它相对于客户端应用程序的当前工作目录。

LogFile (LOG) 连接参数是连接特定的，因此，您可以从单个应用程序为不同的连接设置不同的 LogFile 参数。

典型的日志文件的内容如下：

```
Mon Aug 28 2006 12:29:46 12:29:46 正在试图连接，连接时使用的
是:UID=DBA;PWD=*****;DBF='C:\Documents and Settings\All Users\Documents
\SQL Anywhere 11\Samples\demo.db'; ENG=demoll;START='C:\Program Files\SQL
Anywhere 11\bin32\dbeng11.exe';CON='Sybase Central 1';ASTOP=YES;LOG=c:
\mylog.txt 12:29:46 正在试图连接到正在运行的服务器...12:29:46 正在尝试启动
SharedMemory 链接...

12:29:46 链接已成功启动

12:29:46 正在尝试 SharedMemory 连接 (没有 sasrv.ini 高速缓存地址)

12:29:46 未能通过 SharedMemory 进行连接

12:29:46 未找到服务器，正在试图运行 START 行...12:29:47 已自动启动服务器，正在试图连接，
连接时使用的是:UID=DBA;PWD=*****;DBF='C:\Documents and Settings\All Users
\Documents\SQL Anywhere 11\Samples\demo.db';ENG=demoll;START='C:\Program
Files\SQL Anywhere 11\bin32\dbeng11.exe';CON='Sybase Central 1';ASTOP=YES
12:29:47 正在尝试 SharedMemory 连接 (没有 sasrv.ini 高速缓存地址)

12:29:47 已通过 SharedMemory 连接到服务器

12:29:47 已连接到 SQL Anywhere 服务器版本 11.0.0.2456 12:29:47 应用程序信息:
12:29:47 IP=10.25.99.227;HOST=mymachine-XP;OS='Windows XP Build 2600 Service
Pack 2';PID=0x21c;THREAD=0xa38;EXE='C:\Program Files\SQL Anywhere
11\bin32\scjview.exe';VERSION=11.0.0.2456;
API=iAnywhereJDBC;TIMEZONEADJUSTMENT=-240 12:29:47 已连接到服务器，正在尝试连接到当
前运行的数据库...12:29:48 [ 1] 已成功连接到数据库 12:29:53 [ 1] 由于预读缓冲区的
限制，预取的行数量 12:29:53 [ 1] 已减至 168。请考虑使用 PrefetchBuffer 连接参数。
```

另请参见

- “连接参数”一节第 80 页
- “解决连接参数冲突”一节第 81 页
- “LogFile 协议选项 [LOG]”一节第 297 页

示例

以下命令行可启动 Interactive SQL，并使用 LogFile (LOG) 连接参数连接到示例数据库：

```
dbisql -c "DSN=SQL Anywhere 11 Demo;LOG=d:\logs\test.txt"
```

NewPassword 连接参数 [NEWPWD]

让用户可以更改口令（即使口令已经到期），而无需 DBA 干预。

用法

任何地方。只有 Microsoft Windows 上支持提示输入新口令的客户端库。

值

字符串、*

缺省值

不更改口令，客户端库不提示输入新口令。

注释

当您使用 `password_life_time` 或 `password_expiry_on_next_login` 选项实施登录策略时，此连接参数非常有用。也可以通过使 `login_procedure` 发出 [口令已到期] 出错信号来实施口令到期策略。

如果用户提供新口令，数据库服务器就会验证用户 ID 和口令，并在调用 `login_procedure` 选项前尝试更改该口令。此过程允许用户更改到期口令，而不涉及 DBA。如果设置了 `verify_password_function` 选项，则会验证新口令。如果通过集成登录或 Kerberos 登录进行验证，则不校验原始口令，数据库服务器会忽略新口令值，口令也不会被更改。

在 Microsoft Windows 上，如果使用特殊值 *，则只有在现有口令过期时客户端库才会在尝试连接期间提示输入新口令。用户必须提供现有口令和新口令，并确认新口令。用户填写完各字段并单击 [确定] 后，系统会验证旧口令，数据库服务器也会尝试更改该口令。如果设置了 `verify_password_function` 选项，则会验证新口令。验证用户口令是否过期的过程、系统提示输入口令、验证和更改口令都通过单一连接调用客户端库进行。

如果所处环境不支持口令提示，用户会收到一条 [口令已到期] 错误消息。在 Microsoft Windows 环境中，如果调用应用程序具有多个顶级窗口或者该应用程序的顶级窗口为最小化，提示窗口可能不会成功避免与调用应用程序窗口（它可能不是对话框模式或者具有相应的父窗口）的相互作用。

在 Windows 环境下，如果使用 ODBC `SQLDriverConnect` 函数且 `DriverCompletion` 参数为 `SQL_DRIVER_NOPROMPT` 外的任意值，则当口令过期时，该连接就会提示输入新口令。如果 `DBPROP_INIT_PROMPT` 属性为 `DBPROMPT_NOPROMPT` 外的任意值，则在 OLE DB 中连接可能会提示输入新口令。以上两种情况的作用方式都类似于指定了 `NewPassword=*` 连接参数。

另请参见

- “GRANT 语句”一节 《SQL Anywhere 服务器 - SQL 参考》
- “`login_procedure` 选项 [数据库]”一节第 513 页
- “`verify_password_function` 选项 [数据库]”一节第 560 页
- “`post_login_procedure` 选项 [数据库]”一节第 531 页

示例

以下连接字符串用于更改用户 Test1 在连接时的口令：

```
"UID=Test1;PWD=welcome;NEWPWD=hello"
```

在 Windows 环境下，以下连接字符串用于在现有口令到期时提示用户 Test1 输入新口令：

```
"UID=Test1;PWD=welcome;NEWPWD=*"
```

Password 连接参数 [PWD]

为连接提供口令。

用法

任何位置

值

字符串

缺省值

不提供口令。

注释

数据库的每个用户都有口令。必须提供口令，用户才可以连接数据库。口令的最大长度为 255 字节并且区分大小写。口令中不能包含前导空格、尾随空格或分号。

密码 (PWD) 连接参数不加密。如果将口令存储在数据源中，则应当使用 EncryptedPassword (ENP) 连接参数。Sybase Central 和 SQL Anywhere ODBC 配置工具均使用加密口令。

如果同时指定 Password (PWD) 连接参数和 EncryptedPassword (ENP) 连接参数，则 Password (PWD) 连接参数优先。

您也可以可以在 [连接] 窗口和 [SQL Anywhere 11 的 ODBC 配置] 窗口中的 [口令] 文本框中设置此参数。

小心

在 DSN 或文本文件中存储口令存在很大的安全风险。数据库中存储有任何敏感数据时，建议不要这么做。Sybase Central 和 SQL Anywhere ODBC 配置工具都使用加密口令在 DSN 中存储口令，但是加密口令的安全级别也很低。

另请参见

- “设置口令”一节第 431 页
- “提高口令安全性”一节第 1006 页
- “EncryptedPassword 连接参数 [ENP]”一节第 264 页
- “GRANT 语句”一节 《SQL Anywhere 服务器 - SQL 参考》
- “区分大小写”一节 《SQL Anywhere 服务器 - SQL 的用法》
- “连接参数”一节第 80 页
- “解决连接参数冲突”一节第 81 页

示例

以下连接字符串片段提供用户 ID DBA 和口令 sql。

```
UID=DBA;PWD=sql
```

PrefetchBuffer 连接参数 [PBUF]

为缓冲行设置最大内存量（以字节为单位）。

用法

任何位置

值

Integer [k | m]

缺省值

512 KB (524288)，除 Windows Mobile 之外的所有平台

64 KB (65536 字节)，Windows Mobile

注释

PrefetchBuffer (PBUF) 连接参数控制客户端上所分配的用于存储预读行的内存量。该值以字节为单位，但可使用 **k** 或 **m** 将单位分别指定为千字节或兆字节。此连接参数接受介于 64 KB 和 8 MB 之间的值。

在某些情况下，增加预取行数可以提高查询性能。可使用 PrefetchRows (PROWS) 和 PrefetchBuffer (PBUF) 连接参数增加预取行数。

增加 PrefetchBuffer (PBUF) 连接参数也会增加用于缓冲 GET DATA 请求的内存量。对于一些处理许多 GET DATA (SQLGetData) 请求的应用程序来说，这可能会改善性能。

为与先前的版本兼容，如果指定的值小于 16384，则将其解释为千字节。

建议不要在 PrefetchBuffer 连接参数中使用没有 k 后缀的千字节。请参见“[PrefetchRows 连接参数 \[PROWS\]](#)”一节第 278 页。

另请参见

- “连接参数”一节第 80 页
- “解决连接参数冲突”一节第 81 页

示例

以下连接字符串片段可用于确定 PrefetchBuffer 内存限制是否会减少预取行数。

```
...PrefetchRows=100;LogFile=c:\client.txt
```

以下字符串可用于将内存限制增加到 256 KB：

```
...PrefetchRows=100;PrefetchBuffer=256k
```

PreFetchOnOpen 连接参数

如果启用此参数，预读请求会随游标打开请求一起发送。

用法

ODBC

值

YES、NO

缺省值

NO

注释

如果启用此选项，预读请求会随游标打开请求一起发送，从而不必在每次打开游标后发送读取行的网络请求。列必须已被绑定，以便在打开时能进行预读。使用 `PrefetchOnOpen` 时，在打开和首次读取之间重新绑定游标的列会使系统性能降低。

使 ODBC 调用 `SQLExecute` 或 `SQLExecDirect` 来执行查询或具有返回结果集的存储过程会导致游标打开。

在下列情况下，启用此选项可提高系统性能：

- 网络的等待时间较长
- 应用程序发送过多的打开和关闭游标请求

PrefetchRows 连接参数 [PROWS]

设置在查询数据库时要预读的最大行数。

用法

任何位置

值

整数

缺省值

10

对于 ADO.NET 为 200

注释

如果增加客户端从数据库服务器预读的行数，可以改善只读取相对位置 0 或 1（通过单行读取或宽读取）的游标的性能。宽读取包括嵌入式 SQL 数组读取和 ODBC 块读取。

在以下情况下会得到改善：

- 应用程序较少使用绝对读取来读取许多行（几百或更多）。
- 应用程序以高速率读取行，客户端和服务端位于同一台计算机上或通过快速网络连接。

- 客户端/服务器通信是通过速度较慢的网络（如拨号链接或广域网）进行的。

预读的行数同时受到 PrefetchRows (PROWS) 连接参数和 PrefetchBuffer (PBUF) 连接参数的限制，后者限制用于存储预读行的内存量。请参见“[PrefetchBuffer 连接参数 \[PBUF\]](#)”一节第 277 页。

可预读的最大行数为 1000。

另请参见

- “[连接参数](#)”一节第 80 页
- “[解决连接参数冲突](#)”一节第 81 页

示例

以下连接字符串片段将预读的行数设置为 100:

```
...PrefetchRows=100;...
```

RetryConnectionTimeout 连接参数 [RetryConnTO]

指示客户端库（dblib、ODBC、ADO 等）在找到服务器之前于指定的期间内不断进行连接尝试。

用法

任何位置

值

整数

缺省值

0

注释

此连接指定的值是超时值（以秒为单位）。它不是重试连接尝试的计数器。缺省值零指出仅应尝试连接一次。在迭代之间有半秒的延迟，并且仅在由于未找到数据库服务器而导致连接尝试失败的情况下才会重试。对于任何其它错误，将立即返回错误消息。如果未找到数据库服务器，尝试连接的时间至少不会小于由 RetryConnectionTimeout 连接参数指定的时间。

请注意，缺省 TCP 超时为 5 秒，因此，如果连接字符串中包含的 RetryConnTO 值少于 5 秒（例如 LINKS=tcp;RetryConnTO=3），则尝试连接仍将需要 5 秒。

另请参见

- “[Timeout 协议选项 \[TO\]](#)”一节第 306 页

示例

以下连接字符串片段通知客户端库继续重试连接尝试至少 5 秒:

```
...RetryConnTO=5;...
```

ServerName 连接参数 [ENG]

指定所要连接的运行中数据库服务器的名称。这是 EngineName 的同义词。

用法

网络服务器或个人服务器。

值

字符串

缺省值

缺省的本地数据库服务器。

注释

当某个数据库服务器启动时，它会尝试成为该计算机上的缺省数据库服务器。在没有缺省服务器时，要启动的第一个数据库服务器会成为缺省数据库服务器。该计算机上未显式指定数据库服务器名称的共享内存连接尝试将连接到缺省服务器。

如果要连接到缺省的本地数据库服务器，则无需 ServerName。

如果有多个本地数据库服务器在运行，或要连接到网络服务器，则必须提供一个 ServerName。在 [连接] 窗口和 [SQL Anywhere 11 的 ODBC 配置] 窗口中，它是 [服务器名] 字段。

如果要自动启动服务器，可使用此参数提供服务器名称。

服务器名按照客户端计算机的字符集进行解释。建议不要在服务器名中使用非 ASCII 字符。

名称必须是有效的标识符。数据库服务器名不能：

- 以空格、单引号或双引号开头
- 以空格结尾
- 含有分号
- 超过 250 个字节

对于 Windows 和 Unix，9.0.2 和更早期版本的客户端不能连接到名称超过以下长度的 10.0.0 和更高版本的数据库服务器：

- 对于 Windows 共享内存，为 40 字节
- 对于 Unix 共享内存，为 31 字节
- 对于 TCP/IP，为 40 字节

注意

建议在所部署应用程序的连接字符串中包含 ServerName 参数。这样可确保当计算机运行多个 SQL Anywhere 数据库服务器时，应用程序可连接到正确的服务器，并且还有助于防止出现与计时相关的连接故障。

建议对正在由已部署的应用程序使用的数据库服务器使用 -xd 选项，并且所有客户端都通过使用 ENG 连接参数来明确指定它们应连接到的数据库服务器的名称。这样可确保当一台计算机上有多个 SQL Anywhere 数据库服务器正在运行时，数据库能够连接到正确的数据库服务器。

另请参见

- “标识符”一节 《SQL Anywhere 服务器 - SQL 参考》
- “-n 服务器选项”一节第 195 页
- “-xd 服务器选项”一节第 223 页
- “连接参数”一节第 80 页
- “解决连接参数冲突”一节第 81 页
- “连接到嵌入式数据库”一节第 118 页

示例

连接到名为 Guelph 的服务器：

```
ENG=Guelph
```

StartLine 连接参数 [START]

启动从应用程序运行的个人数据库服务器。

用法

嵌入式数据库

值

字符串

缺省值

无 StartLine 参数。

注释

只有在您要连接当前未在运行的数据库服务器时，才应当提供 StartLine (START) 连接参数。StartLine 连接参数是用于启动个人数据库服务器的命令行。如果 CommLinks [LINKS] 参数包含 TCPIP，数据库服务器将不会自动启动。

注意

如果要指定数据库名、数据库文件或服务器，建议使用 DBN、DBF 和 ENG 连接参数而不使用 StartLine 连接参数。

以下命令使用了建议的语法：

```
START=dbeng11 -c 8M;ENG=mydb;DBN=mydb;DBF=c:\sample.db
```

建议不要使用以下语法：

```
START=dbeng11 -c 8M -n mydb "c:\sample.db"
```

有关可用选项的详细信息，请参见“SQL Anywhere 数据库服务器”一节第 148 页。

注意

只有在无法与指定数据库服务器建立连接，或无法在某个已运行的数据库服务器上启动并连接的数据库时，才使用 StartLine 连接参数启动数据库服务器。例如，假设启动一个运行数据库的数据库服务器，如下所示：

```
dbeng11 c:\mydb.db
```

连接另一个数据库（未使用 ENG 连接参数指定数据库服务器名）：

```
dbisql -c "START=dbsrv11 -c 8M;DBN=seconddb;DBF=c:\myseconddb.db;UID=DBA;PWD=sql"
```

在这种情况下，dbsrv11 数据库服务器不会启动。而是使用之前用来启动 *mydb.db* 的 dbeng11 数据库服务器启动并连接 *myseconddb.db*。

但是，如果指定了 ENG=*server-name*，且名为 *server-name* 的数据库服务器未运行，则 dbsrv11 数据库服务器将会启动。

另请参见

- “连接参数”一节第 80 页
- “解决连接参数冲突”一节第 81 页
- “CommLinks 连接参数 [LINKS]”一节第 254 页
- “连接到嵌入式数据库”一节第 118 页

示例

以下数据源片段启动高速缓存为 8 MB 的个人数据库服务器。

```
StartLine=dbeng11 -c 8M;DBF=samples-dir\demo.db
```

有关 *samples-dir* 的信息，请参见“示例目录”一节第 368 页。

Unconditional 连接参数 [UNC]

使用 db_stop_engine 函数可停止数据库服务器，使用 db_stop_database 函数可停止数据库，即使在该数据库服务器尚有连接时也可执行此操作。

用法

仅限 db_stop_engine 和 db_stop_database 函数

值

YES、NO

缺省值

NO

注释

db_stop_engine 和 db_stop_database 函数分别用于关闭数据库服务器或数据库。如果在连接字符串中指定 UNC=YES，即使存在活动连接，也会关闭数据库服务器或数据库。如果未将 Unconditional 设置为 YES，则只有在没有活动连接时才会关闭数据库服务器或数据库。

另请参见

- “db_stop_database 函数”一节 《SQL Anywhere 服务器 - 编程》
- “db_stop_engine 函数”一节 《SQL Anywhere 服务器 - 编程》
- “连接参数”一节第 80 页
- “解决连接参数冲突”一节第 81 页

Userid 连接参数 [UID]

指定用于登录数据库的用户 ID。

用法

任何位置

值

字符串

缺省值

无

注释

除非使用集成登录或 Kerberos 登录，否则在连接到数据库时，必须始终提供用户 ID。

另请参见

- “连接参数”一节第 80 页
- “解决连接参数冲突”一节第 81 页
- “数据库权限和特权概述”一节第 422 页

示例

以下连接字符串片段提供用户 ID DBA 和口令 sql:

```
UID=DBA;PWD=sql
```

网络协议选项

网络协议选项（包括客户端的和服务器的）使您能够避开不同网络协议实现的缺陷。

您可以在服务器命令中提供网络协议选项。例如：

```
dbsrv11 -x tcpip(PARM1=value1;PARM2=value2;...)
```

从客户端将协议选项作为 CommLinks (LINKS) 连接参数输入：

```
CommLinks=tcpip(PARM1=value1;PARM2=value2;...)
```

如果参数中有空格，必须用引号将网络协议选项括起来，系统命令解释器才能正确进行分析：

```
dbsrv11 -x "tcpip(PARM1=value1;PARM2=value2;...)"
CommLinks="tcpip(PARM1=value1;PARM2=value2;...)"
```

在 Unix 下，如果提供多个参数，也需要使用引号，因为 Unix 将分号解释为命令分隔符。

布尔参数用 YES、Y、ON、TRUE、T 或 1 打开，用 NO、N、OFF、FALSE、F 和 0 关闭。这些参数不区分大小写。

所提供的示例应当全部在一行上输入；您还可以将它们放在配置文件中并使用 @ 服务器选项调用配置文件。

TCP/IP、HTTP 和 HTTPS 协议选项

TCP/IP、HTTP 和 HTTPS 可使用以下选项。

TCP/IP	HTTP 和 HTTPS
“Broadcast 协议选项 [BCAST]” 一节第 285 页	“DatabaseName 协议选项 [DBN]” 一节第 291 页
“BroadcastListener 协议选项 [BLISTENER]” 一节第 286 页	“Identity 协议选项” 一节第 294 页
“ClientPort 协议选项 [CPORT]” 一节第 290 页	“Identity_Password 协议选项” 一节第 294 页
“DoBroadcast 协议选项 [DOBROAD]” 一节第 291 页	“KeepaliveTimeout 协议选项 [KTO]” 一节第 295 页
“Host 协议选项 [IP]” 一节第 292 页	“LocalOnly 协议选项 [LOCAL]” 一节第 296 页
“LDAP 协议选项 [LDAP]” 一节第 295 页	“LogFile 协议选项 [LOG]” 一节第 297 页
“LocalOnly 协议选项 [LOCAL]” 一节第 296 页	“LogFormat 协议选项 [LF]” 一节第 297 页

TCP/IP	HTTP 和 HTTPS
“MyIP 协议选项 [ME]” 一节第 301 页	“LogMaxSize 协议选项 [LSIZE]” 一节第 298 页
“ReceiveBufferSize 协议选项 [RCVBUFSZ]” 一节第 302 页	“LogOptions 协议选项 [LOPT]” 一节第 299 页
“SendBufferSize 协议选项 [SNDBUFSZ]” 一节第 303 页	“MaxConnections 协议选项 [MAXCONN]” 一节第 300 页
“ServerPort 协议选项 [PORT]” 一节第 303 页	“MaxRequestSize 协议选项 [MAXSIZE]” 一节第 301 页
“TDS 协议选项” 一节第 305 页	“MyIP 协议选项 [ME]” 一节第 301 页
“Timeout 协议选项 [TO]” 一节第 306 页	“ServerPort 协议选项 [PORT]” 一节第 303 页
“VerifyServerName 协议选项 [VERIFY]” 一节第 308 页	“Timeout 协议选项 [TO]” 一节第 306 页

Broadcast 协议选项 [BCAST]

指定用来发送广播消息的 IP 地址。

用法

TCP/IP

值

字符串（以 IP 地址的形式）

缺省值

广播到同一子网上的所有地址。

注释

缺省广播地址由本地 IP 地址和子网掩码生成。子网掩码指示哪部分 IP 地址识别网络，哪部分识别主机。

例如，子网为 10.24.98.x，掩码为 255.255.255.0，则缺省广播地址为 10.24.98.255。

在 Windows 平台上指定 IPv6 地址时，应使用接口标识符。Unix 平台在 IPv6 地址中支持接口标识符和接口名称。而在 Linux（内核 2.6.13 及更高版本）上，则需要使用接口标识符。请参见“[SQL Anywhere 中的 IPv6 支持](#)”一节第 135 页。

另请参见

- [“BroadcastListener 协议选项 \[BLISTENER\]” 一节第 286 页](#)
- [“DoBroadcast 协议选项 \[DOBROAD\]” 一节第 291 页](#)
- [“使用广播转发器实用程序查找数据库服务器” 一节第 128 页](#)

示例

以下连接字符串示例通知客户端使用 IPv6 时仅在 2 号接口上广播：

```
LINKS=tcPIP (BROADCAST=ff02::1%2)
```

BroadcastListener 协议选项 [BLISTENER]

控制指定端口的广播监听。

用法

TCP/IP（服务器端）

值

YES、NO

缺省值

YES

注释

该选项允许您针对该端口关闭广播监听。

使用 **-sb 0** 与在 TCP/IP 上指定 BroadcastListener=NO 等效。

如果广播监听处于关闭状态，则数据库服务器不会对 UDP 广播作出响应。这意味着，客户端必须使用 HOST= TCP 协议选项指定数据库服务器的主机名，或者向 LDAP 注册数据库服务器并使用客户端上的 LDAP 查找数据库服务器。这也表示 dblocate 实用程序的输出中不包括数据库服务器。

另请参见

- [“-sb 服务器选项” 一节第 206 页](#)
- [“Broadcast 协议选项 \[BCAST\]” 一节第 285 页](#)
- [“DoBroadcast 协议选项 \[DOBROAD\]” 一节第 291 页](#)

示例

启动接受 TCP/IP 连接的数据库服务器，并要求 TCP/IP 连接使用 Host 协议选项：

```
dbSRV11 -x tcPIP (BroadcastListener=NO) ...
```

以下客户端连接字符串片段用于连接到数据库服务器：

```
...LINKS=tcPIP;HOST=myserver;...
```


certificate_company 协议选项

只有当证书的 [组织] 字段与此值相匹配时，才强制客户端接受服务器证书。

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》。

语法

`certificate_company=organization`

用法

TLS、HTTPS

缺省值

无

注释

SQL Anywhere 客户端信任所有由证书颁发机构签名的证书，因此它们也可能信任同一证书颁发机构发放给其它公司的证书。如果没有方法予以区分，客户端可能会将竞争对手的数据库服务器误当成自己的服务器，并无意中将敏感信息发送给该服务器。此选项指定了更深一层的校验，即证书的标识部分中的 [组织] 字段也要与您指定的值匹配。

只有在 Web 服务客户端过程中才支持 HTTPS。请参见“CREATE PROCEDURE 语句（Web 服务）”一节《SQL Anywhere 服务器 - SQL 参考》。

另请参见

- “certificate_name 协议选项”一节第 288 页
- “certificate_unit 协议选项”一节第 289 页
- “trusted_certificates 协议选项”一节第 307 页
- “Encryption 连接参数 [ENC]”一节第 265 页
- “加密 SQL Anywhere 客户端/服务器通信”一节第 1042 页
- “证书创建实用程序 (createcert)”一节第 705 页

示例

以下命令使用传送层安全将 SQL Anywhere 示例数据库连接到 Interactive SQL。

```
dbisql -c
"UID=DBA;PWD=sql;ENG=demo;LINKS=tcPIP;ENC=TLS (
tls_type=RSA;FIPS=n;trusted_certificates=c:\temp\myident;
certificate_unit='SA';certificate_company='Sybase iAnywhere';
certificate_name='Sybase')"
```

certificate_name 协议选项

只有当证书的 [公用名] 字段与此值相匹配时，才强制客户端接受服务器证书。

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》。

语法

`certificate_name=common-name`

用法

TLS、HTTPS

缺省值

无

注释

SQL Anywhere 客户端信任所有由证书颁发机构签名的证书，因此它们也可能信任同一证书颁发机构发放给其它公司的证书。如果没有方法予以区分，客户端可能会将竞争对手的数据库服务器误当成自己的服务器，并无意中将敏感信息发送给该服务器。此选项指定了更深一层的校验，即证书的标识部分中的 [公用名] 字段也要与指定的值匹配。

只有在 Web 服务客户端过程中才支持 HTTPS。请参见“CREATE PROCEDURE 语句（Web 服务）”一节《SQL Anywhere 服务器 - SQL 参考》。

另请参见

- “certificate_company 协议选项”一节第 287 页
- “certificate_unit 协议选项”一节第 289 页
- “trusted_certificates 协议选项”一节第 307 页
- “Encryption 连接参数 [ENC]”一节第 265 页
- “加密 SQL Anywhere 客户端/服务器通信”一节第 1042 页
- “证书创建实用程序 (createcert)”一节第 705 页

示例

以下命令使用传送层安全将 SQL Anywhere 示例数据库连接到 Interactive SQL。

```
dbisql -c
"UID=DBA;PWD=sql;ENG=demo;LINKS=tcPIP;ENC=TLS (
tls_type=RSA;FIPS=n;trusted_certificates=c:\temp\myident;
certificate_unit='SA';certificate_company='Sybase iAnywhere';
certificate_name='Sybase')"
```

certificate_unit 协议选项

只有当证书的 [组织单位] 字段与此值相匹配时，才强制客户端接受服务器证书。

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》。

语法

`certificate_unit=organization-unit`

用法

TLS、HTTPS

缺省值

无

注释

SQL Anywhere 客户端信任所有由证书颁发机构签名的证书，因此它们也可能信任同一证书颁发机构发放给其它公司的证书。如果没有方法予以区分，客户端可能会将竞争对手的数据库服务器误当成自己的服务器，并无意中将敏感信息发送给该服务器。此选项指定了更深一层的校验，即证书的标识部分中的 [组织单位] 字段也要与指定的值匹配。

只有在 Web 服务客户端过程中才支持 HTTPS。请参见“CREATE PROCEDURE 语句（Web 服务）”一节《SQL Anywhere 服务器 - SQL 参考》。

另请参见

- “certificate_company 协议选项”一节第 287 页
- “certificate_name 协议选项”一节第 288 页
- “trusted_certificates 协议选项”一节第 307 页
- “Encryption 连接参数 [ENC]”一节第 265 页
- “加密 SQL Anywhere 客户端/服务器通信”一节第 1042 页
- “证书创建实用程序 (createcert)”一节第 705 页

示例

以下命令使用传送层安全将 SQL Anywhere 示例数据库连接到 Interactive SQL。

```
dbisql -c
"UID=DBA;PWD=sql;ENG=demo;LINKS=tcPIP;ENC=TLS (
tls_type=RSA;FIPS=n;trusted_certificates=c:\temp\myident;
certificate_unit='SA';certificate_company='Sybase iAnywhere';
certificate_name='Sybase')"
```

ClientPort 协议选项 [CPORT]

指定客户端应用程序通过 TCP/IP 进行通信所用的端口号。

用法

TCP/IP（仅限客户端）

值

整数

缺省值

由网络实现为每个连接动态地指派。如果没有防火墙限制，则建议不使用此参数。

注释

此选项用于通过防火墙的连接，因为防火墙软件根据 TCP/UDP 端口进行数据过滤。除非由于防火墙原因需要使用此参数，否则建议不要使用。

ClientPort 选项指定客户端应用程序通过 TCP/IP 进行通信所用的端口号。您可以指定一个端口号，也可以指定单个端口号和端口号范围的组合。例如：

- (cport=1234)
- (cport=1234,1235,1239)
- (cport=1234-1238)
- (cport=1234-1237,1239,1242)

如果要使用给定的数据源或给定的连接字符串建立多个连接，最好指定端口号的列表或范围。如果您指定一个端口号，则应用程序一次只能保持一个连接。实际上，即使是在关闭一个连接之后，也会有几分钟的停顿时期，在此期间无法使用特定的端口建立任何新连接。如果指定端口号的列表和/或范围，应用程序会一直尝试这些端口号，直到找到一个可成功捆绑到的端口号。

另请参见

- “Host 协议选项 [IP]” 一节第 292 页
- “DoBroadcast 协议选项 [DOBROAD]” 一节第 291 页
- “ServerPort 协议选项 [PORT]” 一节第 303 页
- “穿过防火墙进行连接” 一节第 136 页

示例

以下连接字符串片段在使用端口 6000 的应用程序和使用端口 5000 且名为 my-server 的服务器间建立连接：

```
CommLinks=tcip(ClientPort=6000;ServerPort=5000);ServerName=my-server
```

以下连接字符串片段从可使用端口 5050 到 5060 以及端口 5040 和 5070 的应用程序建立连接，实现与使用缺省服务器端口且名为 my-server 的服务器的通信：

```
CommLinks=tcip(ClientPort=5040,5050-5060,5070);  
ServerName=my-server
```

DatabaseName 协议选项 [DBN]

指定处理 Web 请求时要使用的数据库名，或者使用 REQUIRED 或 AUTO 关键字指定是否需要在 URI 中使用数据库名。

用法

HTTP、HTTPS

值

AUTO、REQUIRED、*database-name*

缺省值

AUTO

注释

如果此参数设置为 REQUIRED，则 URI 必须指定数据库名。

如果此参数设置为 AUTO，则 URI 可以指定数据库名，但不必这么做。如果 URI 中不包含数据库名，将使用服务器上的缺省数据库来处理 Web 请求。由于在设置为 AUTO 时服务器必须确定 URI 中是否包含数据库名，因而在 Web 站点设计中应避免多义性。

如果此参数设置为数据库的名称，则使用该数据库处理所有 Web 请求。URI 不得包含数据库名。

示例

以下命令启动两个数据库，但是只允许通过 HTTP 访问其中的一个。

```
dbsrv11 -xs http(DBN=web) samples-dir\demo.db web.db
```

DoBroadcast 协议选项 [DOBROAD]

控制客户端如何搜索数据库服务器，控制数据库服务器在启动时是否进行广播。

用法

TCP/IP

值

ALL、NONE、DIRECT（客户端）

YES、NO（服务器端）

缺省值

ALL（客户端）

YES（服务器端）

注释

客户端用法 如果 DoBroadcast=ALL，会通过执行广播来搜索数据库服务器。广播首先进入本地子网。如果指定了 HOST=，还将广播包发送到每个主机。所有广播包都是 UDP 包。

如果 DoBroadcast=DIRECT，则不会通过对本地子网执行广播来搜索数据库服务器。广播包只发送到列在 HOST (IP) 协议选项中的主机。如果指定 DoBroadcast=DIRECT，则需要 HOST (IP) 协议选项。

指定 DoBroadcast=NONE 会导致不使用任何 UDP 广播，而且会忽略服务器地址高速缓存 (*sasrv.ini*)。TCP/IP 连接直接通过指定的 HOST/PORT 建立，而且系统会验证服务器名。使用 TCP/IP，通过将 VerifyServerName (VERIFY) 协议选项设置为 NO，可以选择不验证服务器名。除非使用 LDAP，否则 HOST (IP) 协议选项为必需参数，而 ServerPort (PORT) 协议选项则为可选项。

对于 DIRECT 和 NONE，则必须用 HOST 选项指定服务器主机。

服务器端用法 设置 DoBroadcast=NO 会阻止数据库服务器在启动时通过广播来查找同名的其它服务器。在极少数情况下此设置非常有用，但通常不建议这样做。

另请参见

- “Broadcast 协议选项 [BCAST]” 一节第 285 页
- “BroadcastListener 协议选项 [BLISTENER]” 一节第 286 页

示例

以下命令将启动客户端，但不会通过广播来搜索数据库服务器。而是只在名为 silver 的计算机上查找服务器。

```
CommLinks=tcPIP (DOBROADCAST=DIRECT;HOST=silver) demo
```

Host 协议选项 [IP]

指定要由客户端库在直近网络外部搜索的其它计算机。

用法

TCP/IP

值

字符串

缺省值

无其它计算机。

注释

HOST 指定要由客户端库在直接网络外部搜索的其它计算机。在服务器上，执行该搜索以避免使用重复名称启动服务器。在 HOST 协议中指定主机并不意味着数据库服务器必须在指定的主机上运行。

对于 TCP/IP，该地址可为 *hostname* IP 地址。您也可以指定一个 PORT 值。

在 Windows 平台上指定 IPv6 地址时，应使用接口标识符。Unix 平台在 IPv6 地址中支持接口标识符和接口名称。而在 Linux（内核 2.6.13 及更高版本）上，则需要使用接口标识符。请参见“SQL Anywhere 中的 IPv6 支持”一节第 135 页。

如果使用 `-z` 选项，则服务器在启动过程中会将寻址信息显示在数据库服务器消息窗口中。此外，如果指定了 `LogFile` 连接参数，则客户端应用程序会将此信息写入其日志文件。

您可以使用以逗号分隔的地址列表来搜索多台计算机。还可以将端口号附加到 IP 地址，并将冒号用作分隔符。或者，也可以显式地指定主机和服务器端口，如 `HOST=myhost;PORT=5000`。对于 IPv6 地址，必须用圆括号将该地址括起来，例如 `(fe80::5445:5245:444f):2638`。

要为单个参数指定多个值，请使用以逗号分隔的列表。当您指定多个端口和服务器时，可通过在 `HOST (IP)` 协议选项（而非 `PORT` 参数）中指定端口来将特定端口与特定服务器相关联。

`IP` 和 `HOST` 是同义词。

另请参见

- “`ClientPort` 协议选项 [`CPORT`]”一节第 290 页

示例

以下连接字符串片段指示客户端在计算机 `kangaroo` 和 `197.75.209.222`（端口 `2369`）上查找数据库服务器：

```
LINKS=tcpip(IP=kangaroo,197.75.209.222:2369)
```

以下连接字符串片段指示客户端在计算机 `my-server` 和 `kangaroo` 上查找数据库服务器。尝试与在端口 `2639` 上运行的第一个响应的主机建立连接。

```
LINKS=tcpip(HOST=my-server,kangaroo;PORT=2639)
```

以下连接字符串片段指示客户端在 `host1` 上查找运行于端口 `1234` 的服务器，在 `host2` 上查找运行于端口 `4567` 的服务器。客户端不在 `host1` 的端口 `4567` 上查找，也不在 `host2` 的端口 `1234` 上查找。

```
LINKS=tcpip(HOST=host1:1234,host2:4567)
```

以下连接字符串片段指示客户端在 IPv6 地址上查找服务器：

```
LINKS=tcpip(HOST=fe80::5445:5245:444f)
```

以下示例使用 IPv6 地址及 `Host` 协议选项进行说明：

```
Global scope address, unique everywhere, so no interface index is required
// no index required
-c "links=tcpip(Host=fd77:55d:59d9:56a:202:55ff:fe76:df19) "
// all communication is done through interface 2
-c "links=tcpip(Host=fd77:55d:59d9:56a:202:55ff:fe76:df19%2) "
// all communication is done through eth0
-c "links=tcpip(Host=fd77:55d:59d9:56a:202:55ff:fe76:df19%eth0) "

Link scope address, addresses are unique on each interface
// possibly ambiguous (this host may exist through both eth0 and eth1)
-c "links=tcpip(Host=fe80::202:55ff:fe76:df19) "
// not ambiguous because it must use interface 2
-c "links=tcpip(Host=fe80::202:55ff:fe76:df19%2) "
// not ambiguous because it must use eth0
-c "links=tcpip(Host=fe80::202:55ff:fe76:df19%eth0) "
```

Identity 协议选项

指定标识文件的名称。

用法

HTTPS

值

字符串

缺省值

无缺省标识文件名。

注释

此必需选项用于指定标识文件的名称。标识文件包含公共证书及其专有密钥，对于不是自签名的证书，标识文件也包含所有签名证书，其中包括加密证书。该证书的口令必须通过 Identity_Password 参数指定。

另请参见

- [“设置传送层安全”一节第 1035 页](#)
- [“Identity_Password 协议选项”一节第 294 页](#)

示例

启动要求 Web 连接使用特定的加密证书的服务器。

```
dbsrv11 -xs https(Identity=cert.file;Identity_Password=secret) ...
```

Identity_Password 协议选项

指定加密证书的口令。

用法

HTTPS

值

字符串

缺省值

无缺省标识文件口令。

注释

此必需选项用于指定与 Identity 协议选项指定的加密证书相匹配的口令。

另请参见

- “设置传送层安全”一节第 1035 页
- “Identity 协议选项”一节第 294 页

示例

启动要求 Web 连接使用特定的加密证书的服务器。

```
dbsrv11 -xs https(Identity=cert.file;Identity_Password=secret) ...
```

KeepaliveTimeout 协议选项 [KTO]

指定数据库服务器对于完整请求的最长等待时间（以秒为单位）。

用法

HTTP

值

整数

缺省值

60

注释

通常，连接会在每次请求之后被关闭。当客户端请求 Keep-Alive 选项时，HTTP 连接可在每个请求和响应后保持打开状态，从而可以在同一连接上执行多个请求。

连接打开后，客户端将在指定的时间内发送完整的 HTTP 请求，包括 POST 请求的主体。在请求了 Keep-Alive 的连接上，发送完结果后超时值即被重置，这样每个请求在开始时都像刚刚打开一个新连接一样。

如果不希望连接超时，可指定 `kto=0`。

KeepaliveTimeout 和 Timeout 协议选项的区别在于：KeepaliveTimeout 用于指定自连接打开之时起的总时间，而 Timeout 用于指定请求内两个数据包之间的最长时间。

另请参见

- “处理 HTTP 标头”一节 《SQL Anywhere 服务器 - 编程》
- “Timeout 协议选项 [TO]”一节第 306 页

LDAP 协议选项 [LDAP]

允许客户端查找数据库服务器，而无需指定 IP 地址。

用法

TCP/IP

值

YES、NO 或 *filename*

缺省值

YES

缺省文件名为 *saldap.ini*。

注释

通过让数据库服务器自行注册到 LDAP 服务器，允许客户端查询该 LDAP 服务器。这可以让运行在 WAN 上或穿过防火墙的客户端查找服务器，而无需指定 IP 地址。这还使定位实用程序 (dblocate) 能够查找这样的服务器。

指定 LDAP=*filename* 会打开 LDAP 支持，并以指定文件作为配置文件。指定 LDAP=YES 会打开 LDAP 支持并将 *saldap.ini* 用作配置文件。

可以使用文件隐藏实用程序通过简单加密的方式隐藏 *saldap.ini* 文件的内容。请参见“[隐藏 .ini 文件的内容](#)”一节第 724 页。

LDAP 只与 TCP/IP 一起使用。

另请参见

- “[使用 LDAP 服务器进行连接](#)”一节第 138 页

LocalOnly 协议选项 [LOCAL]

允许客户端选择只连接到本地计算机上的服务器（如果存在）。

用法

TCP/IP、HTTP、HTTPS

值

YES、NO

缺省值

NO

注释

如果在本地计算机上找不到具有匹配的服务器名的服务器，服务器将不会自动启动。

只有在指定 DoBroadcast=ALL（缺省值）时，LocalOnly (LOCAL) 协议选项才有用。

如果设置 LocalOnly=YES，则使用常规的广播机制，但是将忽略来自其它计算机上的服务器的广播响应。

可以对服务器使用 LocalOnly (LOCAL) 协议选项，以将连接限制为本地计算机。试图从远程计算机连接将无法找到此服务器，并且定位 [dblocate] 实用程序将无法看到此服务器。运行 LocalOnly

(LOCAL) 协议选项被设置为 YES 的服务器，可以将网络服务器作为个人服务器运行，而不受连接或 CPU 限制。

另请参见

- [“Broadcast 协议选项 \[BCAST\]”一节第 285 页](#)
- [“启动监听 Web 请求的数据库服务器”一节《SQL Anywhere 服务器 - 编程》](#)

LogFile 协议选项 [LOG]

指定数据库服务器用来记录 Web 请求信息的文件的名称。

用法

HTTP、HTTPS

值

Filename

缺省值

无

注释

指定数据库服务器将有关 Web 请求的信息写入到的文件的名称。

另请参见

- [“LogFormat 协议选项 \[LF\]”一节第 297 页](#)
- [“LogMaxSize 协议选项 \[LSIZE\]”一节第 298 页](#)
- [“LogOptions 协议选项 \[LOPT\]”一节第 299 页](#)

LogFormat 协议选项 [LF]

控制写入日志文件的消息的格式及消息中显示的字段。

用法

HTTP、HTTPS

值

Format-string

缺省值

@T - @W - @I - @P - "@M @U @V" - @R - @L - @E

注释

此参数控制写入日志文件的消息的格式以及消息中显示的字段。如果消息以字符串形式显示，则在写入每条消息时将替换以下代码的当前值。

- **@@** @ 字符。
- **@B** 开始处理请求的日期和时间（由于错误而无法排队的请求除外）。
- **@C** 客户端连接的日期和时间。
- **@D** 与请求相关的数据库的名称。
- **@E** 错误消息的文本（如果发生错误）。
- **@F** 请求处理结束的日期和时间。
- **@I** 客户端的 IP 地址。
- **@L** 包括消息标头和消息主体在内的响应的长度（以字节为单位）。
- **@M** HTTP 请求方式。
- **@P** 与请求相关的监听器端口。
- **@Q** 请求排队等候处理的日期和时间（由于错误而无法排队的请求除外）。
- **@R** HTTP 响应的状态代码和说明。
- **@S** HTTP 状态代码。
- **@T** 当前日志条目写入的日期和时间。
- **@U** 请求的 URI。
- **@V** 请求的 HTTP 版本。
- **@W** 用于处理请求的时间 (@F - @B)，或者在由于出错而没有处理请求时为 0.000。

另请参见

- [“LogFile 协议选项 \[LOG\]” 一节第 297 页](#)
- [“LogMaxSize 协议选项 \[LSIZE\]” 一节第 298 页](#)
- [“LogOptions 协议选项 \[LOPT\]” 一节第 299 页](#)

LogMaxSize 协议选项 [LSIZE]

控制数据库服务器在其中写入 Web 请求信息的日志文件的最大大小。

用法

HTTP、HTTPS

值

Integer [k | m | g]

缺省值

0

注释

当日志文件达到指定的大小时，将重命名该文件并创建另一个日志文件。如果 `LogMaxSize` 为 0，则日志文件大小没有限制。缺省值以字节为单位，但可使用 **k**、**m** 或 **g** 将单位分别指定为千字节、兆字节或千兆字节。

另请参见

- [“LogFile 协议选项 \[LOG\]”一节第 297 页](#)
- [“LogFormat 协议选项 \[LF\]”一节第 297 页](#)
- [“LogOptions 协议选项 \[LOPT\]”一节第 299 页](#)

LogOptions 协议选项 [LOPT]

指定数据库服务器用来写入 Web 请求信息的日志中所记录的消息类型。

用法

HTTP、HTTPS

值NONE、OK、INFO、ERRORS、ALL、*status-codes*、REQHDRS、RESHDRS、HEADERS**缺省值**

ALL

注释

可用的值包括选择特殊消息类型的关键字和 HTTP 状态代码。可以指定多个值，各值间以逗号分隔。

以下关键字控制记录的消息类别：

- **NONE** 不记录任何内容。
- **OK** 记录成功完成的请求（20x HTTP 状态代码）。
- **INFO** 记录返回结束或未修改的状态代码的请求（30x HTTP 状态代码）。
- **ERRORS** 记录所有错误（40x 和 50x HTTP 状态代码）。
- **ALL** 记录所有请求。

此外，还有以下通用 HTTP 状态代码可供使用。它们可用于记录返回特殊状态代码的请求：

- **C200** 确定
- **C400** 错误请求

- **C401** 未授权
- **C403** 禁止
- **C404** 未找到
- **C408** 请求超时
- **C501** 未实现
- **C503** 服务不可用

此外，以下关键字可用于获取有关记录的消息的详细信息：

- **REQHDRS** 记录请求时，同时将请求的消息头写入日志文件。
- **RESHDRS** 记录请求时，同时将响应的消息头写入日志文件。
- **HEADERS** 记录请求时，同时将请求和响应的消息头写入日志文件（与REQHDRS,RESHDRS 相同）。

另请参见

- [“LogFile 协议选项 \[LOG\]”一节第 297 页](#)
- [“LogFormat 协议选项 \[LF\]”一节第 297 页](#)
- [“LogMaxSize 协议选项 \[LSIZE\]”一节第 298 页](#)

MaxConnections 协议选项 [MAXCONN]

指定数据库服务器所接受的同时发生的连接数。

用法

HTTP、HTTPS

值

大小

缺省值

5（个人服务器）

已授权连接的数目（网络服务器）

注释

服务器所接受的同时发生的连接数。值 0 表示无限制。

另请参见

- [“MaxRequestSize 协议选项 \[MAXSIZE\]”一节第 301 页](#)

MaxRequestSize 协议选项 [MAXSIZE]

指定数据库服务器可接受的最大请求的大小。

用法

HTTP、HTTPS

值

Integer [k | m | g]

缺省值

100k

注释

服务器接受的最大请求的大小。缺省值以字节为单位，但可使用 **k**、**m** 或 **g** 将单位分别指定为千字节、兆字节或千兆字节。如果请求的大小超出此限制，将关闭连接，同时向客户端返回消息 [413 请求实体太大]。该值仅限制请求的大小，并不限制响应的大小。值 0 禁用该限制，但应慎用。无此限制时，恶意的客户端可能造成服务器过载，或者导致服务器内存不足。

另请参见

- [“MaxConnections 协议选项 \[MAXCONN\]” 一节第 300 页](#)

示例

以下命令行（全部在一行中输入）指示服务器接受大小不超过 150000 字节的请求：

```
dbsrv11 -xs http{MaxRequestSize=150000}
```

MyIP 协议选项 [ME]

控制客户端是否尝试确定寻址信息。

用法

TCP/IP、HTTP、HTTPS

值

字符串

注释

MyIP (ME) 协议选项专为具有多个网络适配器的计算机提供。

每个适配器都有一个 IP 地址。在缺省情况下，数据库服务器使用它找到的每一个网络接口。如果不希望数据库服务器监听所有网络接口，则可指定要在 MyIP (ME) 协议选项中使用的每个接口的地址。

如果将关键字 NONE 作为 IP 号提供，则不尝试确定寻址信息。在有些计算机上执行此操作成本会比较 高，例如，有多块网卡的计算机或者装有远程访问 (RAS) 软件和一块网卡的计算机，关键字 NONE 就是提供给此类计算机上的客户端使用的。此选项不在服务器上使用。

请用逗号分隔多个 IP 地址。

在 Windows 平台上指定 IPv6 地址时，应使用接口标识符。而在 Unix 平台上指定 IPv6 地址时，既可以使用接口标识符，也可以使用接口名称。而在 Linux（内核 2.6.13 及更高版本）上，则需要使用接口标识符。请参见“SQL Anywhere 中的 IPv6 支持”一节第 135 页。

另请参见

- “使用 TCP/IP 协议”一节第 135 页

示例

以下命令行（全部在一行中输入）指示服务器使用两块网卡。

```
dbsrv11 -x tcpip(MyIP=192.75.209.12,192.75.209.32) "samples-dir\demo.db"
```

以下命令行（在一行中输入所有内容）指示数据库服务器使用 IPv6 网卡：

```
dbsrv11 -x tcpip(MyIP=fe80::5445:5245:444f) "samples-dir\demo.db"
```

有关 *samples-dir* 的信息，请参见“示例目录”一节第 368 页。

以下连接字符串片段指示客户端不尝试确定寻址信息。

```
LINKS=tcpip(MyIP=NONE)
```

ReceiveBufferSize 协议选项 [RCVBUFSZ]

设置 TCP/IP 协议栈所用缓冲区的大小。

用法

TCP/IP

值

Integer [k | m | g]

缺省值

与计算机相关。

注释

如果网络上的 BLOB 性能非常重要，您最好增大此值。缺省情况下，所指定缓冲区大小的单位为字节。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。

另请参见

- “使用 TCP/IP 协议”一节第 135 页

SendBufferSize 协议选项 [SNDBUFSZ]

设置 TCP/IP 协议栈所用缓冲区的大小。

用法

TCP/IP

值

Integer [k | m | g]

缺省值

与计算机相关。

注释

缺省值以字节为单位，但可使用 **k**、**m** 或 **g** 将单位分别指定为千字节、兆字节或千兆字节。如果网络上的 BLOB 性能很重要，您最好增加该值。

另请参见

- “使用 TCP/IP 协议” 一节第 135 页

ServerPort 协议选项 [PORT]

指定运行数据库服务器的端口。

用法

TCP/IP、HTTP、HTTPS

值

整数

缺省值

TCP/IP 的缺省值为 2638。HTTP 的缺省值为 80。HTTPS 的缺省值为 443。

注释

Internet 编号授权委员会已经将端口号 2638 指派用于 SQL Anywhere 数据库服务器的 TCP/IP 通信。但是，并不禁止其它应用程序使用这个保留端口，而且这可能会导致在数据库服务器和其它应用程序之间发生寻址冲突。

对于数据库服务器，ServerPort 协议选项指定使用 TCP/IP 在其上进行通信的端口号。您可以指定一个端口号，也可以指定单个端口号和端口号范围的组合。例如：

- (port=1234)
- (port=1234,1235,1239)
- (port=1234-1238)
- (port=1234-1237,1239,1242)

如果指定端口号的列表和/或范围，数据库服务器会尝试绑定到所有指定的端口号。

数据库服务器在大多数操作系统上总是在 UDP 端口 2638 上进行监听，即使您使用网络协议选项指定一个不同的端口时也是如此。应用程序可在不指定端口号的情况下连接到数据库服务器。使该端口可用允许 SQL Anywhere 客户端穿过防火墙查找运行于其它子网上的 SQL Anywhere 数据库服务器。

对于客户端，ServerPort 协议选项会将数据库服务器正在监听 TCP/IP 通信的端口通知客户端。客户端向由 ServerPort (PORT) 协议选项指定的每个端口进行广播以查找该服务器。

如果使用 Web 服务器，缺省情况下，数据库服务器监听标准的 HTTP 和 HTTPS 端口（分别为 80 和 443）。

如果使用 TCP/IP 端口号 2638（缺省值）启动数据库服务器，则服务器还会监听 UDP 端口 2638。数据库服务器监听 UDP 端口并响应这些端口的请求，以便客户端可以通过服务器名找到该数据库服务器。

如果数据库服务器的 TCP/IP 端口号不是 2638，则服务器监听与 TCP/IP 端口相同的 UDP 端口。

数据库服务器为响应客户端广播而发送的 UDP 数据包不包含敏感信息。这些数据包中所含的数据仅限于下列项：

- 数据库服务器名称
- 端口号
- 数据库服务器版本
- 运行于数据库服务器上的数据库的名称

您可以使用 `-dh` 选项对广播请求隐藏数据库名称。还可以指定 `-sb 0` 以完全禁用 UDP 监听器。

在 Mac OS X 上的差别

Mac OS X 不允许在同一 UDP 端口绑定多个进程。如果数据库服务器在其中一个平台上运行，则只监听指定的 UDP 端口，如果未指定端口则监听端口 2638。

也就是说，如果服务器未使用缺省端口 (2638)，则客户端必须指定 TCP/IP 端口号。

例如，如果使用命令 `dbsrv11 -n MyServer samples-dir/demo.db` 启动数据库服务器，则位于同一子网的客户端可使用以下连接参数 `ENG=MyServer;LINKS=tcpip` 来查找该服务器。如果使用以下命令 `dbsrv11 -n SecondServer -x tcpip(PORT=7777) samples-dir/demo.db` 在 Mac OS X 上启动另一个服务器，则位于同一子网的客户端可使用连接参数 `ENG=SecondServer;LINKS=tcpip(PORT=7777)` 来查找该服务器。请注意，如果数据库服务器在非 Mac OS X 的平台上运行，则客户端无需指定 PORT 参数。

此外，在 Mac OS X 上，如果已有一个 SQL Anywhere 数据库服务器在使用端口 2638，而这时又在未指定 PORT 协议选项的情况下启动另一个网络数据库服务器，则第二个网络服务器将无法启动。原因是用户需要知道并在其连接参数中指定服务器的端口号。即使端口 2638 正在使用，个人服务器也可成功启动，因为通常使用共享内存来连接个人服务器。

另请参见

- “-x 服务器选项” 一节第 221 页
- “-xs 服务器选项” 一节第 224 页
- “-sb 服务器选项” 一节第 206 页

示例

以下示例显示如何使用 PORT 协议选项来指定服务器的启动端口。

1. 启动网络数据库服务器：

```
dbsrv11 -x tcpip -n server1
```

现在采用端口号 2638。

2. 尝试启动另一台数据库服务器：

```
dbsrv11 -x tcpip -n server2
```

缺省端口当前已分配，因此服务器会在另一个端口上启动。在 Mac OS X 上，此操作将失败。

3. 如果计算机上的另一个 Web 服务器已在使用端口 80，或者您没有权限在此低端口号上启动服务器，您可能希望启动服务器监听替代端口，例如 8080：

```
dbsrv11 -xs http(port=8080) -n server3 web.db
```

TDS 协议选项

控制是否允许与数据库服务器建立 TDS 连接。

用法

TCP/IP（仅限服务器端）

值

YES、NO

缺省值

YES

注释

要禁止建立到数据库服务器的 TDS 连接，请将 TDS 设置为 NO。如果您希望确保针对服务器只建立加密连接，则此协议选项是禁止建立 TDS 连接的唯一方法。

另请参见

- “-ec 服务器选项” 一节第 171 页

示例

以下命令使用 TCP/IP 协议启动数据库服务器，但是禁止从 Open Client 或 jConnect 应用程序建立连接。

```
dbsrv11 -x tcpip(TDS=NO) ...
```

Timeout 协议选项 [TO]

指定在建立通信时等待响应的长度（以秒为单位）。

用法

TCP/IP、HTTP、HTTPS

值

整数（以秒为单位）

缺省值

对于 TCP/IP，该值为 5。

对于 HTTP 和 HTTPS，该值为 30。

注释

Timeout 还指定在断开连接时等待响应的长度。当您在建立 TCP/IP 通信时遇到问题时，可能希望尝试等待更长的时间。

在数据库服务器上，这是在发送了查找同名服务器的广播后等待的时间。它仅用于服务器的启动，而不会影响客户端的连接。

在服务器上使用 HTTP 或 HTTPS 时，此参数指定接收请求时所允许的最大空闲时间。当到达该限制时，会关闭连接并向客户端返回 [408 请求超时]。值 0 禁用空闲超时，但应慎用。无此限制时，恶意的客户端可能会消耗服务器资源并阻止其它客户端连接。

另请参见

- “KeepaliveTimeout 协议选项 [KTO]” 一节第 295 页

示例

以下数据源片段只启动 TCP/IP 通信链接，超时周期为 20 秒。

```
...  
CommLinks=tcpip(TO=20)  
...
```

trusted_certificates 协议选项

指定包含一个或多个受信任证书的文件的名称和文件名。

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》。

语法

```
trusted_certificates=public-certificate
```

用法

TLS、HTTPS

缺省值

无

注释

客户端使用 `trusted_certificates` 加密协议选项指定受信任的数据库服务器证书。受信任证书可以是服务器的自签名证书、公共企业根证书或属于商业证书颁发机构的证书。如果您使用经 FIPS 认可的 RSA 加密，则必须使用 RSA 生成证书。如果在 Encryption 连接参数中指定了 TLS，则需要此协议选项。

只有在 Web 服务客户端过程中才支持 HTTPS。请参见“CREATE PROCEDURE 语句（Web 服务）”一节《SQL Anywhere 服务器 - SQL 参考》。

另请参见

- “certificate_company 协议选项” 一节第 287 页
- “certificate_name 协议选项” 一节第 288 页
- “certificate_unit 协议选项” 一节第 289 页
- “Encryption 连接参数 [ENC]” 一节第 265 页
- “加密 SQL Anywhere 客户端/服务器通信” 一节第 1042 页
- “证书创建实用程序 (createcert)” 一节第 705 页

示例

以下命令使用传送层安全将 SQL Anywhere 示例数据库连接到 Interactive SQL。

```
dbisql -c
"UID=DBA;PWD=sql;ENG=demo;LINKS=tcpip;ENC=TLS (
tls_type=RSA;FIPS=n;trusted_certificates=c:\temp\myident;
certificate_unit='SA';certificate_company='Sybase iAnywhere';
certificate_name='Sybase')"
```

VerifyServerName 协议选项 [VERIFY]

控制客户端在连接前是否必须校验数据库服务器的名称。

用法

TCP/IP（仅限客户端）

值

YES、NO

缺省值

YES

注释

使用 DoBroadcast=NONE 参数通过 TCP 建立连接时，客户端会建立 TCP 连接，然后验证找到的服务器的名称与要查找的服务器的名称是否相同。如果指定 VerifyServerName=NO，则跳过对服务器名称的验证。这允许 SQL Anywhere 客户端在只知道一个 IP 地址/端口时连接到 SQL Anywhere 服务器。

服务器名仍必须在连接字符串中指定，但会被忽略。只有在指定 DoBroadcast=NONE 时，才使用 VerifyServerName (VERIFY) 协议选项。

如果服务器使用 -sb 0 或 BroadcastListener=NO，则客户端无需指定 DoBroadcast=NONE 就可以连接该服务器，尽管客户端仍必须指定 HOST=。dblocate 实用程序将找不到该服务器。

注意

建议只在不能为每台服务器都提供唯一服务器名并使用该唯一名称进行连接的极少数情况下，才使用此参数。最好的连接方法依然是：为每台服务器提供一个唯一的服务器名，并使用该名称连接该服务器。

另请参见

- [“DoBroadcast 协议选项 \[DOBROAD\]”一节第 291 页](#)

SQL Anywhere for Windows Mobile

目录

在 Windows Mobile 设备上安装 SQL Anywhere	310
使用 Windows Mobile 示例应用程序	313
连接到在 Windows Mobile 设备上运行的数据库	318
配置 Windows Mobile 数据库	321
在 Windows Mobile 上运行数据库服务器	329
在 Windows Mobile 上使用管理实用程序	330
Windows Mobile 上的 SQL Anywhere 功能支持	336

在 Windows Mobile 设备上安装 SQL Anywhere

要求

- Microsoft ActiveSync 3.5 或更高版本。
- SQL Anywhere 支持的 Windows Mobile 设备。
有关 SQL Anywhere 支持的 Windows Mobile 设备的列表，请参见 <http://www.sybase.com/detail?id=1062617>。
- 运行受支持的 Windows 操作系统的计算机。

Windows Mobile 文件位置

SQL Anywhere 在 Windows Mobile 上的安装位置取决于设备的类型和安装位置。不会创建子目录。所有 DLL 都安装在 `\Windows` 目录中。

位置	安装目录
存储卡	<code>\storage-card\Sybase SQL Anywhere 11</code>
主内存	<code>\Program Files\SQLAny11</code>
存储卡	<code>\storage-card\SQLAny11</code>

小心

建议您不要将 SQL Anywhere 安装在存储卡（如 SD 卡）上。

当 Windows Mobile 设备从挂起状态恢复时，所有位于可移动设备上的已打开的文件（包括可执行文件和 DLL 文件）可能已被操作系统关闭。当设备挂起时，操作系统本身无法访问正在运行的程序所使用的可执行文件和 DLL 文件。在这种情况下，操作系统可能会以静默方式从系统中删除该进程（例如 SQL Anywhere 数据库服务器），但不会出现任何错误消息。

安装注意事项：在 Windows Mobile 上使用 ICU

Unicode 归类算法 (UCA) 是对整个 Unicode 字符集进行排序的一种算法。它提供语言方面的正确比较、排序和大小写转换。UCA 作为 Unicode 标准的一部分开发。SQL Anywhere 使用 Unicode 的国际组件 (ICU) 开放源代码库（由 IBM 开发并维护）来实现 UCA。

在 Windows Mobile 上，如果 UCA 用作 NCHAR 归类或 CHAR 归类，则需要 ICU。如果 CHAR 字符集与操作系统字符集不匹配，则在 Windows Mobile 上也需 ICU。

缺省情况下，不会将 ICU 库安装在 Windows Mobile 上，因它会使得在 Windows Mobile 上安装 SQL Anywhere 时安装大小增加大约 1.7 MB。但如果您需要 ICU 库，可以对 SQL Anywhere 安装进行修改。

如果不安装 ICU 库，则在创建数据库时，必须选择其字符集与 Windows Mobile 字符集匹配的归类或 UTF8BIN 归类作为 CHAR 归类。同样，在创建数据库时，必须选择 UTF8BIN 归类作为 NCHAR 归类。

在台式机上创建数据库以部署到 Windows Mobile

当在台式机上创建要部署到 Windows Mobile 设备的数据库时，只能在 Windows Mobile 设备上安装了 ICU 库时才使用 UCA 归类。如果设备上未安装 ICU 库，则在 Windows Mobile 上使用 UCA 的数据库将无法使用。

有关 ICU 的详细信息，请参见“Unicode 归类算法 (UCA)”一节第 393 页和“字符集转换”一节第 386 页。

安装注意事项：在 Windows Mobile 上使用 .NET Compact Framework

虽然 ADO.NET 3.5 是该 API 的最新版本，但是 SQL Anywhere 所支持的大部分设备仅安装了 ADO.NET 1.x 支持。要在设备上使用 ADO.NET 版本 2.0 或 3.5，从 Microsoft 下载 ADO.NET 2.0 和 3.5 的支持程序，并将其安装在设备上。

- **版本 2.0** 有关使用 ADO.NET 2.0 开发应用程序的信息，请参见“SQL Anywhere .NET 数据提供程序”《SQL Anywhere 服务器 - 编程》和“iAnywhere.Data.SQLAnywhere 命名空间 (.NET 2.0)”一节《SQL Anywhere 服务器 - 编程》。
- **版本 3.5** 有关使用 ADO.NET 3.5 开发应用程序的信息，请参见“SQL Anywhere .NET 数据提供程序”《SQL Anywhere 服务器 - 编程》和“iAnywhere.Data.SQLAnywhere 命名空间 (.NET 2.0)”一节《SQL Anywhere 服务器 - 编程》。

有关使用 ADO.NET 的详细信息，请参见“教程：使用 SQL Anywhere .NET 数据提供程序”《SQL Anywhere 服务器 - 编程》。

安装注意事项：Windows Mobile 5.0 for smartphone 的限制

Smartphone 支持所有 SQL Anywhere 服务器 Windows Mobile 功能，但对 Windows Mobile 5.0 有以下限制：

- **不支持共享内存协议** 使用 TCP/IP，即使未指定通信协议。连接时必须始终指定数据库服务器名；如果不指定，则连接失败。
- **不支持 [服务器启动选项] 窗口** 当您启动数据库服务器并且不指定任何选项时，将出现 [服务器启动选项] 窗口。如果尝试启动数据库服务器时提供不完整或不正确的命令，则将显示一条错误，并且不启动数据库服务器。
- **ODBC 连接可能不会提示您提供连接信息** 将 ODBC DriverCompletion 参数用于 SQLDriverConnect 时，系统会提示您提供更多连接信息。不显示此提示。如果 SQLDriverConnect 失败，它不会提示，将返回错误。

- **不支持卸载/重装** 必须在另一个平台上重建 Windows Mobile 数据库，然后将此数据库复制到 Windows Mobile 设备上。这是重建 Windows Mobile 数据库的推荐方法。请参见：
 - “在 Windows Mobile 上重建数据库”一节第 326 页
 - “重建数据库”一节 《SQL Anywhere 服务器 - SQL 的用法》

安装 SQL Anywhere for Windows Mobile

使用以下过程在 Windows Mobile 设备上安装 SQL Anywhere for Windows Mobile。

◆ 安装 SQL Anywhere for Windows Mobile

1. 将 Windows Mobile 设备连接到运行受支持的 Windows 操作系统的计算机上。
2. 选择 [开始] » [程序] » [SQL Anywhere 11] » [部署 SQL Anywhere for Windows Mobile]。
3. 请按照 [SQL Anywhere 部署向导] 中的说明进行操作。

使用 Windows Mobile 示例应用程序

名为 *demo.db* 的示例数据库在 Windows Mobile 设备上的 *My Documents* 目录中。两个版本示例数据库可用于 Windows Mobile；一个包括 Unicode 的国际组件（International Components for Unicode，简称 ICU）库，一个不包括。SQL Anywhere 使用 ICU 开放源代码库（由 IBM 开发并维护）来实现字符集转换。

有关 ICU 和 Windows Mobile 的详细信息，请参见“[安装注意事项：在 Windows Mobile 上使用 ICU](#)”一节第 310 页。

以下示例应用程序随 SQL Anywhere for Windows Mobile 安装提供：

- ADO.NET 示例
- ESQL 示例
- ODBC 示例
- SQL Anywhere 服务器示例

您可以使用这些应用程序访问示例数据库，并检查 SQL Anywhere for Windows Mobile 的功能。

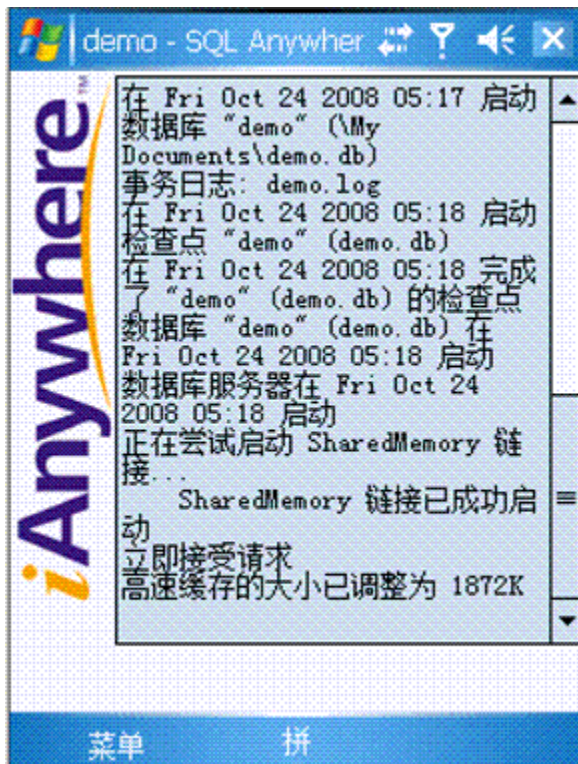
SQL Anywhere 服务器示例

SQL Anywhere 服务器示例使用预设的服务器选项和连接参数，在网络数据库服务器上启动示例数据库。

◆ 启动 SQL Anywhere 服务器示例

- 在 Windows Mobile 设备上，单击 [开始] » [程序] » [SQLAny11] » [Server]，浏览到 SQL Anywhere 安装目录。

示例数据库在网络数据库服务器上开始运行。启动后，数据库服务器以图标形式出现在设备上 [今天] 屏幕的右下角。您可以通过单击此图标查看数据库服务器消息窗口。



现在，您可以从计算机连接到在 Windows Mobile 设备上运行的示例数据库。
使用完示例数据库之后，必须关闭数据库服务器。

◆ 关闭数据库服务器

1. 单击位于 [今天] 屏幕右下角的网络数据库服务器图标。
2. 在菜单上，单击 [关闭]。

ADO.NET 示例

要使用 ADO.NET 示例，必须在设备上安装 Microsoft .NET Compact Framework 版本 2.0 或 3.0。Microsoft .NET Compact Framework 版本 2.0 随 Windows Mobile 6 设备提供，但不随 Windows Mobile 5 设备提供。ADO.NET 示例只支持带触摸屏的 Windows Mobile Classic 和 Professional 设备。

您可以从 Microsoft Download Center 下载此组件，网址为 <http://www.microsoft.com/downloads/search.aspx?displaylang=en>。

ADO.NET 示例是一个使用 ADO.NET 编程接口的简单应用程序。可通过此应用程序启动网络数据库服务器上运行的示例数据库，并使用 SQL 语句访问和修改数据。

此示例的源代码位于 `samples-dir\SQLAnywhere\ce\ado_net_sample`。

您可以将此项目从 `samples-dir\SQLAnywhere\ce\ado_net_sample\ado_net_sample.sln` 装载到 Visual Studio 中。

注意

在 [ADO.NET Sample] 用户界面中，SQL 语句必须在同一行中输入。

◆ 使用 ADO.NET 示例

1. 单击 [开始] » [程序] » [SQLAny11] » [ADO.NET Sample]。
2. 单击 [Connect]。
3. 单击 [Exec SQL] 执行缺省 SQL 语句 **SELECT * FROM Employees**。
Employees 表中的数据出现在数据窗口中。
4. 使用数据窗口侧面和底部的滚动条，浏览 Employees 表中的数据。
5. 键入以下查询以访问范围更具体的数据：

```
SELECT EmployeeID, Surname FROM Employees;
```
6. 单击 [Exec SQL] 执行此 SQL 语句。
指定范围的数据会替换数据窗格中的数据。
7. 键入 **SELECT * FROM Employees ORDER BY EmployeeID**，然后单击 [Exec SQL]。
请注意雇员 Matthew Cobb，其 EmployeeID 为 105。
8. 键入 **UPDATE Employees SET Surname = 'Jones' WHERE Surname = 'Cobb'**，然后单击 [Exec SQL] 执行 SQL 语句。
9. 键入 **SELECT * FROM Employees ORDER BY EmployeeID**，然后单击 [Exec SQL]。
请注意，Matthew 的姓氏已经从 Cobb 更改为 Jones。
10. 键入 **UPDATE Employees SET Surname = 'Cobb' WHERE Surname = 'Jones'**，然后单击 [Exec SQL] 撤消对示例数据库所做的更改。
11. 键入 **SELECT * FROM Employees ORDER BY EmployeeID**，然后单击 [Exec SQL] 验证是否已撤消更改。
请注意，Matthew 的姓氏已经更改回 Cobb。
12. 键入 **SELECT * FROM Customers**，然后单击 [Exec SQL] 访问另一个表的数据。
来自 Customers 表的所有数据即会出现在数据窗口中，它们替换了来自 Employees 表的数据。
13. 单击 [Disconnect]，关闭数据库服务器。
[ADO.NET Sample] 即会断开连接，并自动关闭数据库服务器。
14. 单击窗口右上角的 [x]，关闭 [ADO.NET Sample]。

ESQL 示例

ESQL 示例是一个使用嵌入式 SQL 编程接口的简单应用程序。可通过此应用程序启动网络数据库服务器上运行的示例数据库，并能使用 SQL 语句访问数据。

此示例的源代码位于 *samples-dir\SQLAnywhere\ce\esql_sample*。

您可以将此项目文件通过以下位置装载到 Visual Studio 2005 中：*samples-dir\SQLAnywhere\ce\esql_sample\esql_sample.sln*。

注意

在 [ESQL Sample] 用户界面中，SQL 语句必须在同一行中输入。

◆ 使用 ESQL 示例

1. 单击 [开始] » [程序] » [SQLAny11] » [ESQL Sample]，启动 [ESQL Sample]。
2. 单击 [Connect]，使用缺省连接字符串连接到示例数据库。
3. 单击 [ExecSQL] 执行缺省 SQL 语句 **SELECT * FROM Employees**。
Employees 表中的数据出现在数据窗口中。
4. 使用滚动条查看 Employee 表数据。
5. 要访问 Customers 表中的数据，键入 **SELECT * FROM Customers**，然后单击 [ExecSQL]。
在数据窗口中，Customer 数据替换 Employee 数据。
6. 单击 [Disconnect] 关闭网络数据库服务器。
[ESQL Sample] 即会断开连接，并关闭网络数据库服务器。
7. 单击窗口右上角的 [x]，关闭 [ESQL Sample]。

ODBC 示例

ODBC 示例是一个使用 ODBC 编程接口的简单应用程序。可通过此应用程序启动网络数据库服务器上运行的示例数据库，并能使用基本 SQL 语句访问数据。

此示例的源代码位于 *samples-dir\SQLAnywhere\ce\odbc_sample*。

您可以将此项目文件通过以下位置装载到 Visual Studio 2005 中：*samples-dir\SQLAnywhere\ce\odbc_sample\odbc_sample.sln*。

注意

在 [ODBC Sample] 用户界面中，SQL 语句必须在同一行中输入。

◆ 使用 ODBC 示例

1. 单击 [开始] » [程序] » [SQLAny11] » [ODBC Sample]，启动 [ODBC Sample]。

2. 单击 [**Connect**]。
3. 单击 [**ExecSQL**] 执行缺省 SQL 语句 **SELECT * FROM Employees**。
Employees 表中的数据出现在数据窗口中。
4. 使用滚动条查看 Employee 表数据。
5. 要访问 Customers 表中的数据，键入 **SELECT * FROM Customers**，然后单击 [**ExecSQL**]。
在数据窗口中，Customer 数据替换 Employee 数据。
6. 单击 [**Disconnect**] 关闭网络数据库服务器。
[ODBC Sample] 即会断开连接，并关闭网络数据库服务器。
7. 关闭 [ESQL Sample]。

连接到在 Windows Mobile 设备上运行的数据库

如果要将计算机上运行的应用程序连接到 Windows Mobile 设备上运行的数据库，您可以利用计算机与 Windows Mobile 设备之间的 ActiveSync 链接，通过 TCP/IP 进行连接。这样您就可以使用计算机上的管理实用程序来管理 Windows Mobile 数据库。

另请参见

- “在 Windows Mobile 上使用管理实用程序”一节第 330 页

在 Windows Mobile 设备上启动数据库服务器

如果要从台式计算机连接到在 Windows Mobile 上运行的数据库服务器，则在启动服务器时，必须选择 TCP/IP 选项。

◆ 启动 Windows Mobile 设备上用于远程连接的数据库服务器

1. 从 [开始] 菜单单击 [程序] » [SQLAny11]。
2. 单击 [服务器]。
3. 在 [数据库] 字段中键入要启动的数据库文件的名称，或单击 [浏览] 找到要启动的数据库。
缺省情况下，示例数据库位于 `\My Documents\demo.db` 中。
4. 在 [服务器名] 字段中键入要使用的数据库服务器名。
示例数据库服务器的缺省名称为 **demo**。
5. 选择 [使用 TCP/IP]。
要从计算机连接到 Windows Mobile 设备上运行的数据库，必须使用 TCP/IP 连接。
6. 单击 [确定] 启动在网络数据库服务器上运行的示例数据库。

现在，您可以创建 ODBC 数据源以从计算机连接到 Windows Mobile 设备。

另请参见

- “创建 ODBC 数据源以连接到 Windows Mobile 设备”一节第 319 页

确定 Windows Mobile 设备的 IP 地址

当连接到 Windows Mobile 上运行的数据库时，您可能需要 IP 地址来建立连接。

◆ 确定 Windows Mobile 设备的 IP 地址

1. 从 [开始] 菜单单击 [服务器]。
2. 在 [数据库] 字段，键入 `\My Documents\demo.db` 或单击 [浏览] 找到该数据库。

3. 在 [服务器名] 字段中键入要使用的服务器名。
4. 选择 [使用 TCP/IP]。
要从计算机连接到 Windows Mobile 设备上运行的数据库，必须使用 TCP/IP 连接。
5. 在 [选项] 字段中键入 **-z**。
利用 **-z** 选项，服务器在启动过程中输出其 IP 地址。如果断开 Windows Mobile 设备与网络的连接然后重新连接，则地址可能会发生变化。
有关详细信息，请参见“[-z 服务器选项](#)”一节第 226 页。
6. 单击 [确定] 启动在网络数据库服务器上运行的示例数据库。
7. 浏览至设备上的 [今天] 屏幕。
8. 单击位于屏幕右下角的 [数据库服务器] 图标。
IP 地址出现在数据库服务器消息窗口中。

现在，您可以创建 ODBC 数据源以从计算机连接到 Windows Mobile 设备。

有关详细信息，请参见“[创建 ODBC 数据源以连接到 Windows Mobile 设备](#)”一节第 319 页。

创建 ODBC 数据源以连接到 Windows Mobile 设备

本节介绍如何在 Windows 计算机上创建 ODBC 数据源，以连接到 Windows Mobile 设备上运行的数据库。

有关 ODBC 数据源的详细信息，请参见“[创建 ODBC 数据源](#)”一节第 90 页。

◆ 创建 ODBC 数据源以连接到 Windows Mobile 设备

1. 在 Windows 台式计算机上，选择 [开始] » [程序] » [SQL Anywhere 11] » [ODBC 管理器]。
2. 在 [用户 DSN] 选项卡上，单击 [添加]。
3. 选择 [SQL Anywhere 11]，然后单击 [完成]。
4. 在 [ODBC] 选项卡上，在 [数据源名称] 字段中键入 **MobileServer**。
5. 单击 [登录] 选项卡，选择 [提供用户 ID 和口令]。确保 [用户 ID] 和 [口令] 字段为空。
每次连接到数据库时，您都必须提供用户 ID 和口令。
6. 单击 [网络] 选项卡，选择 [TCP/IP]，然后键入连接参数。

例如，键入 **Host=169.254.2.1;Port=2638;DoBroadcast=none**。

- **Host** 指定 Windows Mobile 设备监听的 IP 地址。

如果使用 USB 连接连接 Windows Mobile 设备，请使用缺省 IP 地址 **169.254.2.1**。

有关详细信息，请参见“[在 Windows Mobile 设备上启动数据库服务器](#)”一节第 318 页。

- **Port** 指定 Windows Mobile 设备监听的端口号。此参数是可选的。

缺省代理端口为 **2638**。

- **DoBroadcast** 控制进行 TCP/IP 连接的方式。此参数为可选参数。

如果指定 **DoBroadcast=none**，则直接使用指定的端口进行 TCP/IP 连接。如果已创建可连接到 Windows Mobile 设备的代理端口，则应使用此设置。

如果指定 **DoBroadcast=direct**，则不会通过对本地子网执行广播来搜索数据库服务器。而是需要提供主机 IP 地址。

有关详细信息，请参见 [“DoBroadcast 协议选项 \[DOBROAD\]”](#) 一节第 291 页。

7. 单击 **[确定]**。

现在，您可以使用此数据源从计算机连接到 Windows Mobile 设备上运行的数据库。

有关详细信息，请参见 [“在 Windows Mobile 上使用管理实用程序”](#) 一节第 330 页。

配置 Windows Mobile 数据库

SQL Anywhere 完整版本中可用的大部分 SQL 功能在 Windows Mobile 版本中也可使用。这些功能包括事务处理、参照完整性操作、过程和触发器等等。不过，在 Windows Mobile 上没有 Java 功能和远程数据访问功能。

在对用于 Windows Mobile 设备的数据库设置数据库属性时，您应注意这些不支持的功能。

有关不支持的功能的完整列表，请参见“[Windows Mobile 上的 SQL Anywhere 功能支持](#)”一节第 336 页。

在数据库创建期间会配置以下设置。一旦设置，更改这些属性设置就只能通过重建数据库。

- **区分大小写或不区分大小写** 请参见“[区分大小写](#)”一节《[SQL Anywhere 服务器 - SQL 的用法](#)》。
- **比较时尾随空白的处理方式** 缺省情况下，创建数据库时尾随空白被归类为额外字符。例如，'Dirk' 和 'Dirk ' 不同。您可以创建具有空格填充的数据库，以使尾随空白被忽略不计。请参见“[在比较中忽略尾随空白](#)”一节《[SQL Anywhere 服务器 - SQL 的用法](#)》。
- **页面大小** 请参见“[表大小和页面大小](#)”一节《[SQL Anywhere 服务器 - SQL 的用法](#)》。
- **归类序列和字符集** 在为 Windows Mobile 创建数据库时，您应该使用一个这样的归类：它与 Windows 为该语言使用的字符集相同，无论是单字节还是多字节字符集。例如，如果您正在使用英语、法语或德语，则使用 1252Latin1 归类。如果您正在使用日语，则使用 932JPN 归类；如果您正在使用朝鲜语，则使用 949KOR 归类。请参见“[了解归类](#)”一节第 392 页。

注意

创建在 Windows Mobile 上使用的数据库时，不要在定制选项字符串中指定 locale 或 sorttype。如果指定了这两项，则在 Windows Mobile 设备上可能无法启动。有关归类定制选项的详细信息，请参见“[归类定制选项](#)”一节第 396 页。

由于不支持在 Windows Mobile 上进行字符集转换，因此 Windows Mobile 数据库必须使用操作系统字符集或者 UTF-8。

创建 Windows Mobile 数据库时，必须选择是否安装 ICU 库。请参见“[安装注意事项：在 Windows Mobile 上使用 ICU](#)”一节第 310 页。

在 Windows Mobile 上使用事务日志

事务日志按照发生的时间顺序对数据库所做的所有更改进行存储。如果数据库文件出现介质故障，则事务日志对进行数据库恢复至关重要。它还可以使您的工作更加有效率。缺省情况下，事务日志与数据库文件位于同一目录。在 Windows Mobile 设备上首次启动数据库时，事务日志就会被创建。

在将现有数据库复制到 Windows Mobile 设备时，可以同时复制数据库文件和事务日志文件。如果不将事务日志文件复制到设备，则在 Windows Mobile 设备上启动数据库时，就会创建新的事务日志。新事务日志不会包含原有事务日志中包含的信息。如果上次使用数据库后没有正确关闭，或者数据库涉及到同步，则可能出现这个问题。因此最佳做法是将数据库文件和事务日志文件都复制到 Windows Mobile 设备。

另请参见

- [“事务日志”一节第 14 页](#)

在 Windows Mobile 上使用 jConnect

jConnect 是用于 SQL Anywhere 的一个纯 Java JDBC 驱动程序。Sybase Central 和 Interactive SQL 提供了启用 jConnect JDBC 驱动程序的选项，以使 Java 应用程序能访问 SQL Anywhere 数据库。

缺省情况下，对于为 Windows Mobile 创建的数据库，[\[创建数据库向导\]](#) 不启用 jConnect。但如有必要，您可以选择启用 jConnect。

为数据库添加 jConnect 支持会向系统表中添加许多条目。这会增加数据库的大小，更重要的是，即使您不使用任何 jConnect 功能，也需要增加大约 200 KB 的内存以满足数据库的运行要求。

如果您不打算使用 jConnect，并且您运行在一个内存有限的环境中（例如 Windows Mobile），则建议您不要为数据库添加 jConnect 支持。

另请参见

- [“使用 jConnect JDBC 驱动程序”一节 《SQL Anywhere 服务器 - 编程》](#)

在 Windows Mobile 上使用加密

可以使用简单加密或高度加密来保护您的数据库。数据库初始化之后，更改加密设置的唯一方法就是重建整个数据库。

另请参见

- [“加密和解密数据库”一节第 1019 页](#)
- [“保护 Windows Mobile 数据库的安全”一节第 1029 页](#)

创建 Windows Mobile 数据库

可以为 Windows Mobile 设备创建 SQL Anywhere 数据库：

- 使用 Sybase Central [\[创建数据库向导\]](#) 创建可直接复制到 Windows Mobile 设备的数据库。
- 使用初始化实用程序 (dbinit)，创建可手工复制到 Windows Mobile 设备的数据库。
- 使用 Interactive SQL 中的 CREATE DATABASE 语句，创建可手工复制到 Windows Mobile 设备的数据库。

注意

在 Windows Mobile 上运行数据库时，数据库服务器自动打开校验和。在数据库文件损坏时，它能帮助您较早地将其检测出来。

有关创建 Windows Mobile 数据库时需要做出的决定的信息，请参见：

- “在 Windows Mobile 上使用事务日志”一节第 321 页
- “安装注意事项：在 Windows Mobile 上使用 ICU”一节第 310 页
- “安装注意事项：在 Windows Mobile 上使用 .NET Compact Framework”一节第 311 页

使用 Sybase Central 创建 Windows Mobile 数据库

Sybase Central 具有某些便于创建 Windows Mobile 数据库的功能。Sybase Central 将确保达到 Windows Mobile 数据库的要求，并且允许您选择将数据库文件复制到您的设备。

◆ 使用 Sybase Central 创建 Windows Mobile 数据库，并将其直接复制到 Windows Mobile 设备

1. 将 Windows Mobile 设备连接到计算机。
2. 选择 [开始] » [程序] » [SQL Anywhere 11] » [Sybase Central]。
3. 选择 [工具] » [SQL Anywhere 11] » [创建数据库]。
4. 单击 [在这台计算机上创建数据库]。单击 [下一步]。
5. 指定文件名和目录以在计算机中存储数据库文件，然后单击 [下一步]。
6. 选择 [为 Windows Mobile 创建此数据库]，然后单击 [下一步]。
7. 选择 [将数据库复制到 Windows Mobile 设备]，然后单击 [下一步]。
8. 指定数据库文件复制到的 Windows Mobile 目录。缺省位置为主设备目录。

提示

将数据库复制到 Windows Mobile 设备的 *My Documents* 目录，以简化数据库的启动。

使用 [服务器启动选项] 窗口在 Windows Mobile 设备上启动数据库时，只能使用 [浏览] 搜索 *My Documents* 目录中的数据库文件。

如果数据库未存储在 *My Documents* 目录中，则必须在 [服务器启动选项] 窗口的 [数据库] 字段中键入数据库的路径。

您也可以选择 [复制后删除台式机数据库] 选项。

如果您选择不删除计算机副本，则数据库文件的副本将存储在计算机上在第 5 步中指定的目录中。单击 [下一步]。

9. 指定是否要将数据库复制到 Windows Mobile 设备。
10. 指定要保存事务日志文件的目录。单击 [下一步]。
在 Windows Mobile 设备上，首次在网络数据库服务器上启动数据库时，会在数据库文件所在的目录中创建事务日志文件。
11. 指定是否要使用事务日志镜像。单击 [下一步]。

12. 清除 [安装 jConnect 元数据支持] 选项，然后单击 [下一步]。

13. 通过选择适当的选项为数据库设置加密级别，然后单击 [下一步]。

如果选择高度加密，则必须指定加密密钥。建议选择满足以下条件的密钥值：长度至少为 16 个字符，混合大小写以及包含数字、字母和特殊字符。

小心

请务必将密钥的副本存储在一个安全的位置。每次启动或修改数据库时，都需要使用该密钥。丢失密钥将导致数据库完全无法访问，而且也无法对数据库进行恢复。

14. 选择页面大小，然后单击 [下一步]。

15. 在 [指定其它设置] 页面上，选择 [让每个数据库页都包括校验和]，然后单击 [下一步]。

16. 按照向导中的其余说明进行操作，然后单击 [完成] 创建数据库并将其复制到设备上。

随即出现一个窗口，跟踪文件复制到 Windows Mobile 设备的进度。单击 [关闭]。

指定 NCHAR 数据的归类序列

如果不需要 NCHAR UCA 排序，则 NCHAR 归类序列应为 UTF8BIN。这样，数据库服务器就不需要 ICU 库 (*dbicu11.dll* 和 *dbicudt11.dll*)。选择 [使用所提供的以下归类]，然后选择 [UTF8BIN]。

17. 向导将数据库复制到 Windows Mobile 设备后，请验证文件的位置：

从 [Start] 菜单单击 [Programs] » [File Explorer]，然后浏览到要将数据库复制到的 Windows Mobile 目录。

数据库文件会在此处列出。首次在 Windows Mobile 设备上启动数据库之前，不会出现事务日志文件。

使用 dbinit 创建 Windows Mobile 数据库

初始化实用程序 (dbinit) 可用于创建可在 Windows Mobile 上使用的数据库。不过，您不能从此实用程序将数据库直接复制到 Windows Mobile 设备。您必须将使用 dbinit 实用程序创建的数据库手工复制到 Windows Mobile 设备。

◆ 使用 dbinit 实用程序创建数据库

1. 在命令提示符处，浏览到要创建数据库的目录。例如：

```
cd temp
```

2. 通过运行以下命令创建数据库：

```
dbinit -s database-name.db
```

-s 选项可为数据库启用校验和。

提示

您还可以使用初始化实用程序配置数据库属性，如加密和页面大小。请参见“[初始化实用程序 \(dbinit\)](#)”一节第 729 页。

3. 将数据库复制到 Windows Mobile 设备。

有关将数据库复制到 Windows Mobile 设备的详细信息，请参见“[将数据库复制到 Windows Mobile 设备](#)”一节第 325 页。

使用 CREATE DATABASE 语句创建 Windows Mobile 数据库

使用 CREATE DATABASE 语句可以在计算机上的 Interactive SQL 中创建数据库。不过，您不能从此应用程序将数据库直接复制到 Windows Mobile 设备。您必须将数据库手工复制到 Windows Mobile 设备。

◆ 使用 CREATE DATABASE 语句创建数据库

1. 选择 [开始] » [程序] » [SQL Anywhere 11] » [Interactive SQL]。

如果 [连接] 窗口未自动出现，请选择 [SQL] » [连接]。

2. 在 [标识] 选项卡上，单击 [ODBC 数据源名称]，然后在相邻字段中键入 **SQL Anywhere 11 Demo**。

3. 单击 [确定]。

4. 在 Interactive SQL 的 [SQL 语句] 窗格中键入以下语句：

```
CREATE DATABASE 'c:\\temp\\database-name.db'  
TRANSACTION LOG ON  
CHECKSUM ON;
```

提示

您还可以使用 CREATE DATABASE 语句配置数据库属性，例如加密和页面大小。请参见“[CREATE DATABASE 语句](#)”一节《[SQL Anywhere 服务器 - SQL 参考](#)》。

5. 从 [SQL] 菜单中选择 [执行]。

在计算机的 *c:\temp* 目录中创建了数据库和事务日志。

有关将数据库复制到 Windows Mobile 设备的信息，请参见“[将数据库复制到 Windows Mobile 设备](#)”一节第 325 页。

将数据库复制到 Windows Mobile 设备

使用本节中介绍的方法，可将任何现有的 SQL Anywhere 数据库复制到 Windows Mobile 设备。不过，您必须记住，在将数据库复制到 Windows Mobile 设备时，Windows Mobile 上不支持的所有数据库功能均会失效。请参见“[Windows Mobile 上的 SQL Anywhere 功能支持](#)”一节第 336 页。

◆ 将数据库复制到 Windows Mobile 设备

1. 将 Windows Mobile 设备连接到计算机。
2. 在计算机上打开 Windows 资源管理器。
3. 浏览至计算机上包含您要复制的数据库的目录。
4. 右击数据库文件并选择 **[复制]**。
5. 再打开一个 Windows 资源管理器实例。
6. 浏览至 Windows Mobile 设备上您要存储该数据库文件的目录。

提示

使用 **[服务器启动选项]** 窗口在 Windows Mobile 设备上启动数据库时，只能使用 **[浏览]** 搜索 *My Documents* 目录中的数据库文件。

如果数据库未存储在 *My Documents* 目录中，则必须在 **[服务器启动选项]** 窗口的 **[数据库]** 字段中键入数据库的路径。

7. 在 Windows 资源管理器窗口中右击 Windows Mobile 设备所在的空白区域，然后选择 **[粘贴]**。文件即被复制到 Windows Mobile 设备。

在 Windows Mobile 上重建数据库

在 Windows Mobile 上重建数据库时，可采用以下方法：

- 在另一个平台上重建 Windows Mobile 数据库，然后将该数据库复制到 Windows Mobile 设备。这是重建 Windows Mobile 数据库的推荐方法。
- 使用 dbmlsync 重新填充空数据库。
- 使用 dbremote 重新填充空数据库。
- 在 Windows Mobile 设备上使用 dbunload。此方法在智能手机上不可用。

升级 Windows Mobile 数据库时，建议使用前三种方法。但是，如果这些方法对您不适用，则您可在 Windows Mobile 上使用 dbunload。决定在 Windows Mobile 上使用 dbunload 之前，应考虑在 Windows Mobile 上使用 dbunload 产生的以下影响：

- 数据库服务器临时文件的大小（卸载和重装都可能使此文件增至数兆字节）
- dbunload 及相关组件所需的额外空间
- 在 Windows Mobile 设备上具有多个数据库副本的额外开销

因为在 Windows Mobile 设备上运行 dbunload 所需要的资源可能会比一些设备可用的资源更多，所以建议尽可能在其它平台升级数据库。

注意

如果要在 Windows Mobile 设备上运行 dbunload，必须在 [部署 SQL Anywhere 11 for Windows Mobile 向导] 中选择 [卸载/重装支持]。如果在首次安装 SQL Anywhere for Windows Mobile 时未选择该选项，则可通过修改 SQL Anywhere 安装来添加此支持。

在 Windows Mobile 上使用 dbunload 的注意事项

要在 Windows Mobile 设备上使用 dbunload，请确保执行了以下任务：

- 应将以下文件部署到 SQL Anywhere 安装目录（缺省情况下为 `\Program Files\SQLAny11`）：
 - `dbsrv11.exe`
 - `dbunlspt.exe`
 - `dbunload.exe`
 - `dbrunsql.exe`
- 应将以下文件部署到 `\Windows` 目录：
 - `dblgen11.dll`
 - `dblib11.dll`
 - `dbscript11.dll`
 - `dbtool11.dll`
 - `dbusen.dll`
- 应将以下注册表条目字符串值设置为 SQL Anywhere 软件目录：`HKEY_LOCAL_MACHINE\SOFTWARE\Sybase\SQL Anywhere\11.0\Location`。

可将以下步骤嵌入第三方 Windows Mobile 应用程序中，以便自动对最终用户执行该过程。如果选择执行此操作，那么您应考虑使用 `-qc` 和/或 `-q dbunload` 和 `dbrunsql` 选项，或者调用 `dbtool11.dll` 中的 `DBUnload` 函数。

◆ 卸载 Windows Mobile 上的数据库 (dbunload)

1. 在 Windows Mobile 以外的其它平台上，创建一个新的空 SQL Anywhere 11 数据库。
CHAR 归类序列应与现有数据库的相匹配。如果不需要 NCHAR UCA 排序，则 NCHAR 归类序列应为 UTF8BIN。这样，数据库服务器就不需要 ICU 库（`dbicu11.dll` 和 `dbicudt11.dll`）。
2. 将 SQL Anywhere 11 软件以及空 SQL Anywhere 11 数据库文件复制到 Windows Mobile 设备。请参见“在 Windows Mobile 上使用 dbunload 的注意事项”一节第 327 页。
3. 确保在设备上未运行任何数据库服务器。
4. 运行以下命令：

```
dbunload-path\dbunload -c "UID=DBA;PWD=DBA-
password;CHARSET=none;DBF=existing-database" unload-directory
```

5. 确保 dbunload 成功执行，然后关闭 dbunload 窗口。
6. 运行以下命令：

```
dbrunsql-path\dbrunsql -c "UID=DBA;PWD=sql;CHARSET=none;DBF=new-empty-
SQLAnywhere11databasefile" -g- \reload.sql
```

7. 确保 `dbrunsql` 成功执行，然后关闭 `dbrunsql` 窗口。
8. 从 Windows Mobile 设备删除 `reload.sql` 文件和 `unload-directory`。

备份 Windows Mobile 数据库

在出现数据损坏或介质故障的情况下，备份和恢复对于确保不丢失数据是至关重要的。最好将 Windows Mobile 数据库备份到物理上分开的位置，以防止由于 Windows Mobile 设备被盗或丢失，或者 Windows Mobile 设备上的介质故障而导致的数据丢失。

大多数备份和恢复实用程序都可用于 Windows Mobile。不过，这些实用程序并没有太多用处，因为您无法在 Windows Mobile 上使用它们将备份存储到物理上分开的位置。替代方法是通过将整个数据库文件复制到计算机来备份数据。您还可以使用同步在计算机上维护 Windows Mobile 数据库的最新副本。请参见“了解 [MobiLink 同步](#)”《[MobiLink - 入门](#)》。

消除 Windows Mobile 数据库

SQL Anywhere for Windows Mobile 不支持 [\[消除数据库向导\]](#)、`DROP DATABASE` 语句或消除实用程序 (`dberase`)。您必须手工将数据库从 Windows Mobile 设备中消除。尝试删除数据库时，数据库必须处于未运行状态。

有两种方法可将数据库从 Windows Mobile 设备中消除。您可以通过设备界面消除数据库，或者将设备连接到计算机并使用 Windows 资源管理器消除数据库。

删除数据库之后，再删除事务日志文件（如果存在）。

◆ 使用设备界面消除数据库

1. 从 **[Start]** 菜单单击 **[Programs]** » **[File Explorer]**，然后浏览至包含要消除的数据库文件的目录。
2. 单击数据库文件并按住不放。
3. 单击 **[Delete]**。
4. 单击 **[Yes]** 以确认删除。

◆ 使用 Windows 资源管理器消除数据库

1. 将 Windows Mobile 设备放入底座，并确保它通过 ActiveSync 连接到计算机。
2. 在计算机上打开 Windows 资源管理器。
3. 浏览至存储数据库文件的 Windows Mobile 目录。
4. 右击数据库文件并选择 **[删除]**。
5. 单击 **[是]**。

在 Windows Mobile 上运行数据库服务器

客户端/服务器通常会这样安排：运行数据库服务器的计算机要比运行客户端应用程序的计算机功能更强、资源更多。而对于 Windows Mobile 则显然不是这种情况，运行数据库服务器的计算机功能较弱。

为 Windows Mobile 提供了网络数据库服务器。其文件名为 *dbsrv11.exe*。网络数据库服务器支持通过 TCP/IP 进行通信。因为 Windows Mobile 支持网络数据库服务器，所以您可以在计算机上运行管理实用程序来对 Windows Mobile 数据库执行任务。例如：

- 您可以在计算机上使用 Sybase Central 管理数据库。
- 您可以在计算机上使用 Interactive SQL 来装载和卸载数据，以及执行查询。

有关详细信息，请参见“在 Windows Mobile 上使用管理实用程序”一节第 330 页。

除非显式请求 TCP/IP 网络链接，否则 Windows Mobile 数据库服务器不会启动该网络链接。

有关在 Windows Mobile 上启动数据库服务器的详细信息，请参见“教程：从 Sybase Central 运行 Windows Mobile 数据库”一节第 330 页。

在 Windows Mobile 上，在第一个 SQL Anywhere 数据库服务器已运行的情况下尝试启动第二个数据库服务器时，会将第一个服务器推到前台。这是 Windows Mobile 应用程序的标准行为。因此，不能在一个 Windows Mobile 设备上同时运行两个数据库服务器。但是，SQL Anywhere 支持在单个数据库服务器上运行多个数据库。

在 Windows Mobile 上指定服务器选项

在启动数据库服务器时可以指定服务器和数据库的选项，以调整 SQL Anywhere 的行为和性能。您可以从许多选项中进行选择，以指定多种功能，例如用作高速缓存的内存量、在数据库服务器上启动数据库所需的权限级别以及要使用的网络协议。

在 Windows Mobile 上，在 [服务器启动选项] 窗口中指定选项。这与其它 Windows 操作系统不同，在其它 Windows 操作系统上，可在命令行上设置数据库服务器选项。大多数服务器选项均可用于 Windows Mobile。

有关数据库服务器选项的详细信息，请参见“数据库服务器”第 147 页。

有关不支持的选项的信息，请参见“Windows Mobile 上的数据库服务器选项支持”一节第 338 页。

在 Windows Mobile 上使用管理实用程序

本节介绍使用 SQL Anywhere 数据库管理实用程序管理 Windows Mobile 数据库时的具体注意事项。

教程：从 Sybase Central 运行 Windows Mobile 数据库

Sybase Central 是一个数据库管理工具，它提供了管理 SQL Anywhere 的图形用户界面。Sybase Central 也可用于管理其它产品，包括 MobiLink 同步。

完成本教程后，您将能执行与数据库服务器相关的关键任务：启动和停止服务器、在数据库服务器上运行单个和多个数据库，以及连接到数据库。

要求

- 在开始本教程之前，请完成以下部分中的所有任务：
 - “[连接到在 Windows Mobile 设备上运行的数据库](#)”一节第 318 页
 - “[创建 ODBC 数据源以连接到 Windows Mobile 设备](#)”一节第 319 页
- 将 Windows Mobile 设备连接到计算机。

开始之前

您需要创建两个 Windows Mobile 数据库以用于本教程。

◆ 为 Windows Mobile 设备创建数据库

1. 将 Windows Mobile 设备连接到计算机。
2. 选择 [开始] » [程序] » [SQL Anywhere 11] » [Sybase Central]。
3. 选择 [工具] » [SQL Anywhere 11] » [创建数据库]。
4. 请按照 [创建数据库向导] 中的说明进行操作。
5. 在 [Specify A Database] 页面上，单击 [浏览]，然后为数据库文件选择位置。将数据库文件命名为 Alpha。
6. 在 [选择为 Windows Mobile 创建] 页面上，单击 [为 Windows Mobile 创建此数据库]，然后单击 [下一步]。

该向导测试到 Windows Mobile 设备的连接。

7. 在 [选择复制数据库] 页面上，选择 [将数据库复制到 Windows Mobile 设备]。
8. 在 [Windows Mobile 文件名] 字段中，键入 `\\My Documents\Alpha.db`。
9. 选择 [复制后删除台式机数据库]，然后单击 [下一步]。
10. 在 [为 NCHAR 数据指定归类序列] 页上，选择 [使用所提供的以下归类]，然后选择 [UTF8BIN]。

有关详细信息，请参见“[安装注意事项：在 Windows Mobile 上使用 ICU](#)”一节第 310 页。

11. 单击 [完成]。
12. 单击 [关闭]。
13. 重复此过程并创建一个名为 *My Documents\Beta.db* 的数据库。

第 1 课：启动数据库服务器

本节介绍在 Windows Mobile 上运行单个数据库的简单情况。

◆ 在服务器上启动数据库

1. 在 Windows Mobile 设备上，单击 [开始] » [服务器]。
2. 在 [数据库] 字段中键入要启动的数据库文件的名称，或单击 [浏览] 找到 *My Documents* 目录中的 *Alpha.db* 文件。
3. 在 [服务器名] 字段中键入 **MobileServer**。
4. 选择 [使用 TCP/IP]。

要从计算机连接到 Windows Mobile 设备上运行的数据库，必须使用 TCP/IP 连接。在以后的课程中，您将从计算机建立连接。

5. 在 [选项] 字段中键入 **-gd all**。

-gd 选项设置的权限允许任何用户在网络数据库服务器上启动其它数据库。在以后的课程中需要使用此权限。请参见“[-gd 服务器选项](#)”一节第 179 页。

6. 单击 [确定] 启动在网络数据库服务器上运行的 Alpha 数据库。
7. 浏览至设备上的 [今天] 屏幕。
8. 单击位于屏幕右下角的 [数据库服务器] 图标。

当消息 [立即接受请求] 出现在数据库服务器消息窗口中时，您就可以开始学习下一课了。

下一步做什么？

接下来，您将学习如何在 Windows Mobile 上的网络数据库服务器上启动多个数据库。

第 2 课：在 Windows Mobile 数据库服务器上启动多个数据库

在 Windows Mobile 设备上，在第一个 SQL Anywhere 数据库服务器已运行的情况下尝试启动第二个数据库服务器时，会将第一个服务器数据库推到前台。这是 Windows Mobile 应用程序的标准行为。因此，不能在一个 Windows Mobile 设备上同时运行两个数据库服务器。作为运行多个数据库服务器的替代方法，一个服务器可以运行多个数据库。

◆ 从 Sybase Central 连接到数据库

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 [标识] 选项卡，完成以下字段：
 - 用户 ID DBA
 - 口令 sql
4. 单击 [ODBC 数据源名称]。
5. 单击 [浏览]，选择您在“创建 ODBC 数据源以连接到 Windows Mobile 设备”一节第 319 页中创建的 **MobileServer** 数据源。
6. 单击 [数据库] 选项卡，在 [服务器名] 字段中键入 **MobileServer**。
7. 单击 [确定] 连接到在 Windows Mobile 设备上运行的 *Alpha.db* 数据库。
8. 如果无法连接到数据库服务器：
 - 单击 [网络] 选项卡。
 - 单击 [TCP/IP]。
 - 完成 [主机] 和 [端口] 字段。
 - 单击 [确定]。

现在您已启动数据库服务器并连接到 Alpha 数据库，可以在 Windows Mobile 设备上启动其它数据库了。

◆ 在网络数据库服务器上启动另一个数据库

1. 在 Sybase Central 的左窗格中，右击 [MobileServer]，然后选择 [启动数据库]。
2. 在 [数据库文件] 字段中键入 \My Documents\Beta.db。
3. 单击 [确定] 在网络数据库服务器上启动数据库。

数据库即装载到网络数据库服务器上。现在您必须从计算机启动连接。

◆ 连接到第二个数据库

1. 在 Sybase Central 中，选择 [文件] » [连接]。
2. 单击 [标识] 选项卡，完成以下字段：
 - 用户 ID DBA
 - 口令 sql
3. 单击 [数据库] 选项卡，然后填写以下字段：
 - 数据库文件 Beta
 - 服务器名 MobileServer
4. 单击 [确定] 连接到在 Windows Mobile 设备上运行的 Beta 数据库。

现在您就可以使用 Sybase Central 在 Alpha 和 Beta 数据库中查看和处理数据了。

下一步做什么？

接下来，您将学习如何从数据库断开连接并在 Windows Mobile 上关闭数据库服务器。

第 3 课：在 Windows Mobile 上关闭数据库服务器

您必须先停止与计算机的连接，然后才能在 Windows Mobile 设备上关闭网络数据库服务器。

◆ 断开与 Windows Mobile 数据库的连接

1. 在 Sybase Central 中，选择 [连接] » [断开连接]。
2. 选择对应于 Alpha 数据库的连接。
3. 单击 [确定]。
4. 选择 [连接] » [断开连接]。

Beta 数据库即被断开连接。

使用 Sybase Central 断开与 Windows Mobile 数据库的连接后，就可以关闭网络数据库服务器了。

◆ 关闭服务器

1. 在 Windows Mobile 设备上，单击位于 [今天] 屏幕右下角的 [数据库服务器] 图标。
2. 单击 [菜单] » [关闭]。

参考信息

从 Sybase Central 连接到数据库后，便可向数据库的表中添加数据、添加和编辑数据库对象，以及执行其它管理任务。

有关从 Sybase Central 管理数据库的信息，请参见：

- “使用 SQL Anywhere 插件” 一节第 630 页
- “使用数据库对象” 《SQL Anywhere 服务器 - SQL 的用法》

教程：使用 Interactive SQL 管理 Windows Mobile 数据库

Interactive SQL 是一个应用程序，它可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 提供了一个用于输入 SQL 语句的窗格，并且还提供了用于显示查询的处理情况以及结果集的窗格。

本教程简要介绍如何从计算机使用 Interactive SQL 管理 Windows Mobile 设备上的数据库。您将学习如何从 Interactive SQL 连接到 Windows Mobile 设备上的示例数据库。连接后，您就可以使用 Interactive SQL 执行 SQL 语句。

第 1 课：启动示例数据库

必须在 Windows Mobile 设备上运行了示例数据库后，才可从 Interactive SQL 连接到该数据库。

◆ 启动示例数据库

1. 在 Windows Mobile 设备上，单击 [开始] » [服务器]。
2. 在 [数据库] 字段中，键入示例数据库的路径。缺省位置为 **\My Documents\demo.db**。如果将软件安装在不同位置，则使用 [浏览] 按钮查找数据库。
3. 在 [服务器名] 字段中键入 **MobileServer**。
4. 在 [高速缓存] 字段中键入 **5MB**。
Windows Mobile 上的缺省高速缓存大小为 600 KB。不过，建议您使用较大的高速缓存，这有助于提高性能。
5. 选择 [使用 TCP/IP]。本教程假定您使用缺省代理端口号 2638。
6. 单击 [确定] 启动在网络数据库服务器上运行的示例数据库。
7. 浏览至设备上的 [今天] 屏幕。
8. 单击位于屏幕右下角的 [服务器] 图标。

当消息 [立即接受请求] 出现在数据库服务器消息窗口中时，您就可以开始学习下一课了。

下一步做什么？

接下来，您将学习如何从 Interactive SQL 连接到 Windows Mobile 设备上运行的数据库。

第 2 课：启动 Interactive SQL 并连接

示例数据库在 Windows Mobile 设备上运行后，您可以从 Interactive SQL 连接到示例数据库，并可以从计算机查看和管理该数据库。

◆ 从 Interactive SQL 连接到 Windows Mobile 设备上的数据库

1. 在台式计算机上，选择 [开始] » [程序] » [SQL Anywhere 11] » [Interactive SQL]。
2. 单击 [标识] 选项卡，完成以下字段：
 - 用户 ID 键入 **DBA**。
 - 口令 键入 **sql**。
3. 选择 [ODBC 数据源名称]。
4. 单击 [浏览]，选择您在“创建 ODBC 数据源以连接到 Windows Mobile 设备”一节第 319 页中创建的 **MobileServer** 数据源。
5. 单击 [确定]。
6. 单击 [数据库] 选项卡，在 [服务器名] 字段中键入 **MobileServer**。如果无法连接到服务器：
 - 单击 [网络] 选项卡。

- 单击 [TCP/IP]。
 - 完成 [主机] 和 [端口] 字段。
 - 单击 [确定]。
7. 单击 [确定] 连接到在 Windows Mobile 设备上运行的示例数据库。

下一步做什么？

现在您就可以从 Interactive SQL 查看和管理示例数据库中的数据了。

第 3 课：执行对 Windows Mobile 数据库的查询

Interactive SQL 的一个主要用途就是浏览表数据。Interactive SQL 通过向数据库服务器发送请求来检索信息。接着，数据库服务器会查找该信息并将结果返回给 Interactive SQL。

◆ 对 Windows Mobile 数据库执行 SQL 语句

1. 在 [SQL 语句] 窗格中执行以下语句：

```
SELECT * FROM Employees;
```

2. 从 [SQL] 菜单选择 [执行] 执行此语句。

Employees 表中的所有数据即会出现在 [结果] 窗格中。

3. 从 [SQL] 菜单选择 [断开连接] 断开与 Windows Mobile 数据库的连接。

参考信息

从 Interactive SQL 连接到数据库后，便可以查看和处理数据，并可以添加及修改数据库对象。

有关 Interactive SQL、编写查询和使用 SQL 语句的信息，请参见：

- “使用 Interactive SQL” 一节第 636 页
- “查询数据” 《SQL Anywhere 服务器 - SQL 的用法》
- “对查询结果进行汇总、分组和排序” 《SQL Anywhere 服务器 - SQL 的用法》
- “连接：从多个表检索数据” 《SQL Anywhere 服务器 - SQL 的用法》
- “SQL 语句” 《SQL Anywhere 服务器 - SQL 参考》

Windows Mobile 上的 SQL Anywhere 功能支持

本节列出在 Windows Mobile 上不支持或功能已改变的 SQL Anywhere 的组件和功能。在适当的地方，会列出不支持的功能的替代方法。

有关 Windows Mobile 上支持和不支持的组件的详细信息，请参见 <http://www.sybase.com/detail?id=1062623>。

SQL Anywhere 包括多种用于管理数据库的工具。其中包括 Sybase Central、Interactive SQL 和实用程序。所有这些管理工具都无法部署到 Windows Mobile。替代方法是，从连接到 Windows Mobile 设备的基于 Windows 的计算机执行数据库管理。

有关详细信息，请参见“在 Windows Mobile 上使用管理实用程序”一节第 330 页。

组件或功能	注意事项
应用程序分析	为运行于 Windows Mobile 上的数据库创建跟踪会话时，必须使用 [数据库跟踪向导]（不能使用 [应用程序分析向导]）配置跟踪。此外，必须通过 Windows Mobile 设备到台式计算机上的数据库服务器上运行的 Windows Mobile 数据库副本上跟踪数据。无法自动从 Windows Mobile 设备创建跟踪数据库，且无法跟踪 Windows Mobile 设备上的本地数据库。请参见“应用程序分析”一节《SQL Anywhere 服务器 - SQL 的用法》。
数据库镜像	Windows Mobile 上不支持。
外部存储过程	Windows Mobile 上不支持。
iAnywhere JDBC 驱动程序	Windows Mobile 上不支持。可在 Windows Mobile 上使用 jConnect。
数据库中的 Java	Windows Mobile 上不支持。
jConnect	当您为 Windows Mobile 创建数据库时，可启用 jConnect 驱动程序。如果您想要将数据库移至支持 Java 的计算机，此功能将很有用。不过，启用 jConnect 驱动程序会增加数据库大小，更重要的是，会使运行数据库所需的内存增加大约 200 KB。当在内存有限的环境（如 Windows Mobile）中运行数据库时，应考虑这个额外的内存需求。请参见“在 Windows Mobile 上使用 jConnect”一节第 322 页。
Kerberos 验证	Windows Mobile 上不支持。
LDAP 验证	Windows Mobile 上不支持。

组件或功能	注意事项
ODBC 客户端	由于 Windows Mobile 不提供 ODBC 驱动程序管理器或 ODBC 管理器，因此 SQL Anywhere 使用存储于文件中的 ODBC 数据源。请参见“在 Windows Mobile 上使用 ODBC 数据源”一节第 94 页。
Open Client	Windows Mobile 上不支持。
并行备份	Windows Mobile 上不支持。
个人数据库服务器 (dbeng11)	Windows Mobile 上仅支持网络数据库服务器 (dbsrv11)。此可执行文件支持本地连接和通过网络进行客户端/服务器通信。
远程数据访问 (包括目录访问服务器)	Windows Mobile 上不支持。

Windows Mobile 上的 SQL 语句支持

本节介绍 Windows Mobile 不支持的 SQL 语句，以及功能已改变或受限制的 SQL 语句。

有关 SQL 语句的完整列表，请参见“SQL 语句”《SQL Anywhere 服务器 - SQL 参考》。

SQL 语句	注意事项
BACKUP 语句	Windows Mobile 上仅支持 BACKUP DATABASE DIRECTORY 子句。
CREATE DATABASE 语句	CREATE DATABASE 语句可用于在计算机上初始化数据库，该数据库以后可以复制到 Windows Mobile 设备中。请参见“创建 Windows Mobile 数据库”一节第 322 页。
CREATE EVENT 语句	Windows Mobile 上不支持 DiskSpace 事件类型。不过，可以使用此语句定义 GlobalAutoIncrement 事件类型或 ServerIdle 事件类型。
CREATE EXISTING TABLE 语句	Windows Mobile 上不支持。
CREATE EXTERNLOGIN 语句	Windows Mobile 上不支持。
CREATE FUNCTION 语句	可以在 Windows Mobile 上使用 CREATE FUNCTION 语句来创建用户定义的 SQL 函数，以在数据库中使用。注意，在 Windows Mobile 上不支持 EXTERNAL NAME 子句。
CREATE SERVER 语句	Windows Mobile 上不支持。

SQL 语句	注意事项
CREATE TABLE 语句	Windows Mobile 上不支持 CREATE TABLE 语句中用于创建代理表的 AT 子句。
DROP DATABASE 语句	Windows Mobile 上不支持。
DROP SERVER 语句	Windows Mobile 上不支持。
INSTALL JAVA 语句	Windows Mobile 上不支持。
REMOVE JAVA 语句	Windows Mobile 上不支持。
REORGANIZE TABLE 语句	Windows Mobile 上不支持。
RESTORE DATABASE 语句	Windows Mobile 上不支持。
START JAVA 语句	Windows Mobile 上不支持
STOP JAVA 语句	Windows Mobile 上不支持。

Windows Mobile 上的数据库服务器选项支持

本节介绍在 Windows Mobile 上不支持或功能已改变的数据库服务器选项。

选项	注意事项
@data 选项	Windows Mobile 上不支持。
-? 服务器选项	Windows Mobile 上不支持。
-cm 服务器选项	Windows Mobile 上不支持。
-cw 服务器选项	Windows Mobile 上不支持。
-ec 服务器选项	Windows Mobile 上不支持高度通信加密 (TLS)。仅支持 none 和 simple 设置。请参见“ -ec 服务器选项 ”一节第 171 页。
-gb 服务器选项	Windows Mobile 上不支持。
-ge 服务器选项	Windows Mobile 上不支持。
-qi 服务器选项	运行时，网络数据库服务器以图标形式出现在 Windows Mobile 设备的 [今天] 屏幕的右下角。此功能不能被禁用。
-s 服务器选项	Windows Mobile 上不支持。

选项	注意事项
-tmf 服务器选项	Windows Mobile 上不支持。
-tmt 服务器选项	Windows Mobile 上不支持。
-u 服务器选项	Windows Mobile 上不支持。
-ua 服务器选项	Windows Mobile 上不支持。
-uc 服务器选项	Windows Mobile 上不支持。
-ud 服务器选项	Windows Mobile 上不支持。
-uf 服务器选项	Windows Mobile 上不支持。
-ui 服务器选项	Windows Mobile 上不支持。
-ut 服务器选项	Windows Mobile 上不支持。
-ux 服务器选项	Windows Mobile 上不支持。
-xp 服务器选项	Windows Mobile 上不支持。
-ze 服务器选项	Windows Mobile 上不支持。

Windows Mobile 上的 Sybase Central 向导支持

下表列出了在 Windows Mobile 上不支持或功能已改变的 Sybase Central 向导，并在可能时提供替代方法。

向导	注意事项
备份数据库向导	Windows Mobile 不支持档案备份。不支持 [备份数据库向导]。请参见“备份类型”一节第 819 页。 在 Windows Mobile 上您可使用 [创建备份映像向导] 作为替代，该向导分别创建数据库文件和事务日志文件的备份。请参见“使用 [备份数据库向导]”一节第 827 页。
更改日志文件设置向导	Windows Mobile 上不支持。

向导	注意事项
创建数据库向导	如果在运行 Sybase Central 的计算机上安装了 Windows Mobile 服务，则此向导会提供可在 Windows Mobile 上创建数据库的选项。请参见“ 创建 Windows Mobile 数据库 ”一节第 322 页。
创建维护计划向导	以下选项在 Windows Mobile 上不可用： <ul style="list-style-type: none"> ● 完全档案备份 ● 备份到磁带 ● 以电子邮件方式发送维护计划报告
消除数据库向导	Windows Mobile 上不支持。
迁移数据库向导	Windows Mobile 上不支持。
恢复数据库向导	Windows Mobile 上不支持。
创建服务向导	Windows Mobile 上不支持。
翻译日志文件向导	Windows Mobile 上不支持。
卸载数据库向导	此向导不能映射到 Windows Mobile 上存储数据库文件的目录。不过，您可以通过将 Windows Mobile 数据库复制到计算机并使用 [卸载数据库向导] 来卸载该数据库。
升级数据库向导	在 Windows Mobile 不支持此向导。不过，您可以通过将 Windows Mobile 数据库复制到计算机并使用此向导，然后将该数据库复制回 Windows Mobile 设备的方法来升级数据库。请参见“ 升级 SQL Anywhere ”一节《 SQL Anywhere 11 - 更改和升级 》。

Windows Mobile 上的 SQL Remote 支持

Windows Mobile 支持 SQL Remote，但以下情况除外：

组件或功能	注意事项
抽取实用程序 (dbxtract)	Windows Mobile 不支持此实用程序。如有必要，可将 Windows Mobile 数据库复制到计算机，以便使用抽取实用程序。

配置数据库

本节介绍 SQL Anywhere 所使用的文件、数据库限制以及如何配置数据库属性和选项。还将介绍如何配置 SQL Anywhere 安装来处理国际语言问题。

SQL Anywhere 环境变量	343
文件位置和安装设置	367
国际语言和字符集	377
管理用户 ID、特权和权限	415
数据库选项	461
连接属性、数据库属性和数据库服务器属性	565
物理限制	613

SQL Anywhere 环境变量

目录

SQL Anywhere 环境变量简介	344
DYLD_LIBRARY_PATH 环境变量 [Mac OS X]	346
LD_LIBRARY_PATH 环境变量 [Linux 和 Solaris]	347
LIBPATH 环境变量 [AIX]	348
ODBCHOME 环境变量 [Unix]	349
ODBCINI 和 ODBC_INI 环境变量 [Unix]	350
PATH 环境变量	351
SACHARSET 环境变量	352
SADIAGDIR 环境变量	353
SALANG 环境变量	355
SALOGDIR 环境变量	356
SATMP 环境变量	357
SHLIB_PATH 环境变量 [HP-UX]	359
SQLANY11 环境变量	360
SQLANYSAMP11 环境变量	361
SQLCONNECT 环境变量	362
SQLPATH 环境变量	363
SQLREMOTE 环境变量	364
SYBASE 环境变量	365
TMP、TMPDIR 和 TEMP 环境变量	366

SQL Anywhere 环境变量简介

SQL Anywhere 使用环境变量来存储各种类型的信息。并不是在所有情况下都需要设置所有的环境变量。

对于 SQL Anywhere 服务器，可以查看为特定服务器设置的环境变量，方法是使用 `-ze` 选项启动该服务器。请参见“[-ze 服务器选项](#)”一节第 226 页。

在 Windows 上设置环境变量

SQL Anywhere 安装程序会在计算机的属性中创建或修改以下环境变量：PATH 和 SQLANY11。安装 SQL Anywhere 后，必须重新启动计算机才能使这些环境变量生效。

还可以通过修改计算机的属性或是在命令提示符或批处理文件内使用 SET 命令来设置其它环境变量。

在 Mac OS X 中为 Finder 设置环境变量

SQL Anywhere 安装程序会设置以下环境变量：DYLD_LIBRARY_PATH、ODBCINI、PATH 和 SQLANY11。不需要重新启动。

终端会话不会从 Finder 继承环境变量。下节将介绍如何为终端会话设置环境变量。

在 Unix 和 Mac OS X 上设置环境变量

安装 SQL Anywhere 11 后，每个用户均须设置一些环境变量，方可使系统找到并运行 SQL Anywhere 应用程序。为此，SQL Anywhere 安装程序会创建两个文件：`sa_config.sh` 和 `sa_config.csh`。这些文件安装在 `install-dir/bin32` 和 `install-dir/bin64` 中。每个文件均设置了所有需要的用户环境变量。

正如名称所表示的，一个文件要在 Bourne shell (`sh`) 及其派生程序（如 `ksh` 或 `bash`）中使用。另一个文件要在 C-shell (`csh`) 及其派生程序（如 `tcsh`）中使用。

在这些批处理文件中，每个文件都有些语句被注释掉了。根据系统的配置，系统管理员可能需要编辑这些文件并删除注释。

要运行 SQL Anywhere 应用程序，可选择以下几种方式：

1. 如果将环境变量从 `sa_config` 文件添加到系统环境中，则可通过从 GUI（如 X Window 服务器）启动或在终端窗口中键入应用程序名来运行应用程序。
2. 在终端窗口中，如果发起了其中一个 `sa_config` 文件，则可通过键入应用程序的名称来运行该应用程序。
3. `install-dir/bin32s` 和 `install-dir/bin64s` 包含与 SQL Anywhere 应用程序同名的脚本。这些脚本会在启动应用程序前设置合适的环境变量。可以通过运行相应的脚本来运行应用程序。不必在运行这些脚本前发起 `sa_config` 文件。

在 Unix 和 Mac OS X 上发起文件

发起文件是指在 shell 的当前实例中执行文本文件中包含的命令。这是使用 shell 内置的命令实现的。

在 Bourne shell 和其派生程序下，此命令的名称是 `.`（一个英文句点）。例如，如果 SQL Anywhere 安装在 `/opt/sqlanywhere11` 中，应使用以下语句发起 `sa_config.sh`：

```
./opt/sqlanywhere11/bin32/sa_config.sh
```

在 C-shell 和其派生程序下，命令是 `source`。例如，如果 SQL Anywhere 安装在 `/opt/sqlanywhere11` 中，应使用以下语句发起 `sa_config.csh`：

```
source /opt/sqlanywhere11/bin32/sa_config.csh
```

DYLD_LIBRARY_PATH 环境变量 [Mac OS X]

在 Mac OS X 上，该环境变量指定运行时在哪些目录中搜索 SQL Anywhere 应用程序所需的库。

语法

DYLD_LIBRARY_PATH=*path-list*

缺省值

/Applications/SQLAnywhere11/System/lib32

注释

由安装程序创建的 *sa_config.sh* 和 *sa_config.csh* 文件是用于创建或修改上述及其它环境变量的脚本。

另请参见

- [“LD_LIBRARY_PATH 环境变量 \[Linux 和 Solaris\]”](#) 一节第 347 页
- [“LIBPATH 环境变量 \[AIX\]”](#) 一节第 348 页
- [“SHLIB_PATH 环境变量 \[HP-UX\]”](#) 一节第 359 页
- [“在 Unix 和 Mac OS X 上设置环境变量”](#) 一节第 344 页

LD_LIBRARY_PATH 环境变量 [Linux 和 Solaris]

在 Linux 和 Solaris 上，该环境变量指定运行时在哪些目录中搜索 SQL Anywhere 应用程序所需的库。

语法

```
LD_LIBRARY_PATH=path-list
```

缺省值

- `/opt/sqlanywhere11/lib32` (32 位平台)
- `/opt/sqlanywhere11/lib64` (64 位平台)

注释

由安装程序创建的 `sa_config.sh` 和 `sa_config.csh` 文件是用于创建或修改上述及其它环境变量的脚本。

另请参见

- “DYLD_LIBRARY_PATH 环境变量 [Mac OS X]” 一节第 346 页
- “LIBPATH 环境变量 [AIX]” 一节第 348 页
- “SHLIB_PATH 环境变量 [HP-UX]” 一节第 359 页
- “在 Unix 和 Mac OS X 上设置环境变量” 一节第 344 页

LIBPATH 环境变量 [AIX]

在 AIX 上，该环境变量指定运行时在哪些目录中搜索 SQL Anywhere 应用程序所需的库。

语法

LIBPATH=*path-list*

缺省值

- */usr/lpp/sqlanywhere11/lib32* (32 位平台)
- */usr/lpp/sqlanywhere11/lib64* (64 位平台)

注释

由安装程序创建的 *sa_config.sh* 和 *sa_config.csh* 文件是用于创建或修改上述及其它环境变量的脚本。

另请参见

- [“DYLD_LIBRARY_PATH 环境变量 \[Mac OS X\]”](#) 一节第 346 页
- [“LD_LIBRARY_PATH 环境变量 \[Linux 和 Solaris\]”](#) 一节第 347 页
- [“SHLIB_PATH 环境变量 \[HP-UX\]”](#) 一节第 359 页
- [“在 Unix 和 Mac OS X 上设置环境变量”](#) 一节第 344 页

ODBCHOME 环境变量 [Unix]

指定 *.odbc.ini* 文件的位置。

语法

ODBCHOME=*odbc-ini-directory*

注释

.odbc.ini 文件是包含 ODBC 数据源的系统信息文件。如果该文件被命名为 *.odbc.ini* 之外的其它文件名，则必须使用 ODBCINI 或 ODBC_INI 环境变量指定其位置。

有关查找 ODBC 数据源的算法的信息，请参见“在 Unix 上使用 ODBC 数据源”一节第 95 页。

另请参见

- “ODBCINI 和 ODBC_INI 环境变量 [Unix]”一节第 350 页
- “在 Unix 和 Mac OS X 上设置环境变量”一节第 344 页

ODBCINI 和 ODBC_INI 环境变量 [Unix]

指定包含 ODBC 数据源的系统信息文件的路径和名称。

语法

ODBCINI=*odbc-ini-file*

ODBC_INI=*odbc-ini-file*

注释

如果使用这些环境变量中的某一个来指定文件名，则文件名不必为 *.odbc.ini*。提供这两个环境变量是为了与其它产品兼容。

有关查找 ODBC 数据源的算法的信息，请参见“[在 Unix 上使用 ODBC 数据源](#)”一节第 95 页。

另请参见

- “[ODBCHOME 环境变量 \[Unix\]](#)”一节第 349 页
- “[在 Unix 和 Mac OS X 上设置环境变量](#)”一节第 344 页

PATH 环境变量

指定包含 SQL Anywhere 可执行文件的目录位置。

语法

PATH=*path-list*

缺省值

注意
只有安装了相应的组件，才会添加以下路径。

操作系统	缺省位置
Windows (32 位)	<i>C:\Program Files\SQL Anywhere 11\bin32</i>
Windows (64 位)	<i>C:\Program Files\SQL Anywhere 11\bin64</i>
Mac OS X (32 位)	<i>/Applications/SQLAnywhere11/System/bin32</i>
Mac OS X (64 位)	<i>/Applications/SQLAnywhere11/System/bin64</i>
AIX (32 位)	<i>/usr/lpp/sqlanywhere11/bin32</i>
AIX (64 位)	<i>/usr/lpp/sqlanywhere11/bin64</i>
其它 Unix 操作系统 (32 位)	<i>/opt/sqlanywhere11/bin32</i>
其它 Unix 操作系统 (64 位)	<i>/opt/sqlanywhere11/bin64</i>
Linux、Solaris Sparc	<i>/opt/sqlanywhere11/openserver/OCS-15_0/bin</i>

注释

在 Windows 上，安装程序会对 PATH 环境变量进行修改，以加入 SQL Anywhere 可执行文件所在的目录。

在 Unix 上，由安装程序创建的 *sa_config.sh* 和 *sa_config.csh* 文件是用于创建或修改上述及其它环境变量的脚本。

另请参见

- [“在 Windows 上设置环境变量”一节第 344 页](#)
- [“在 Unix 和 Mac OS X 上设置环境变量”一节第 344 页](#)

SACHARSET 环境变量

指定 SQL Anywhere 所使用的字符集。

语法

SACHARSET=*charset*

注释

charset 是字符集名称。

有关推荐的字符集的信息，请参见“建议的字符集和归类”一节第 408 页。

如果未指定 SACHARSET，则字符集根据操作系统而定。

另请参见

- “在 Windows 上设置环境变量”一节第 344 页
- “在 Unix 和 Mac OS X 上设置环境变量”一节第 344 页

SADIAGDIR 环境变量

指定 SQL Anywhere 诊断目录的位置。

语法

SADIAGDIR=*diagnostic-information-directory*

缺省值

操作系统	缺省位置
Windows	<i>%ALLUSERSPROFILE%\Application Data\SQL Anywhere 11\diagnostics</i>
Unix	<i>\$HOME/.sqlanywhere11/diagnostics</i>
Windows Mobile	运行数据库服务器的目录

注释

SQL Anywhere 将崩溃报告和性能统计信息存储在诊断目录中。SADIAGDIR 环境变量用于确定 SQL Anywhere 在其中写入崩溃报告的诊断目录的位置。

如果此环境变量指定的目录不存在，则数据库服务器会如同未设置该环境变量一样操作。

在 Windows（不包括 Windows Mobile）上，诊断信息被写入以下列表中的首个可写目录：

1. 由 SADIAGDIR 环境变量指定的目录。
2. 当前可执行文件的目录。
3. 当前目录。
4. 临时目录。请参见“SATMP 环境变量”一节第 357 页和“TMP、TEMPDIR 和 TEMP 环境变量”一节第 366 页。

在 Windows Mobile 上，诊断信息被写入以下列表中的首个可写目录：

1. 当前可执行文件的目录。
2. 当前目录。
3. 临时目录。请参见“Windows Mobile 上的注册表设置”一节第 374 页。

在 Unix 上，诊断信息被写入以下列表中的首个可写目录：

1. 由 SADIAGDIR 环境变量指定的目录。
2. *\$HOME/.sqlanywhere11/diagnostics* 指定的文件目录。
3. 当前目录。
4. 临时目录。请参见“SATMP 环境变量”一节第 357 页和“TMP、TEMPDIR 和 TEMP 环境变量”一节第 366 页。

注意

在 Unix 上，当数据库或 MobiLink 服务器作为守护程序运行或用户为 root/nobody 时，建议不要将崩溃报告写入用户的主目录。因此，Unix 安装时会提示您选择一个合适的位置，并在 *sa_config.sh* 和 *sa_config.csh* 文件中设置 SADIAGDIR 环境变量。

另请参见

- “支持实用程序 (dbsupport)” 一节第 785 页
- “SQL Anywhere 中的错误报告” 一节第 76 页
- “在 Windows 上设置环境变量” 一节第 344 页
- “在 Unix 和 Mac OS X 上设置环境变量” 一节第 344 页

SALANG 环境变量

指定 SQL Anywhere 所使用的语言代码。

语法

SALANG=*language-code*

注释

language-code 是表示某种语言的双字母组合。例如，设置 **SALANG=DE** 可将缺省语言设置为德语。

有关支持的语言代码的信息，请参见“[了解区域设置语言](#)”一节第 389 页。

以下方法中第一个返回值的方法确定了缺省语言：

1. 检查 SALANG 环境变量。
2. (Windows) 检查安装期间设置的或由 *dblang.exe* 设置的注册表。请参见“[语言选择实用程序 \(dblang\)](#)”一节第 745 页。
3. 查询操作系统以获取语言信息。
4. 如果未设置任何语言信息，则英语将作为缺省语言。

另请参见

- “[语言选择实用程序 \(dblang\)](#)”一节第 745 页
- “[安装时的注册表设置](#)”一节第 374 页
- “[在 Windows 上设置环境变量](#)”一节第 344 页
- “[在 Unix 和 Mac OS X 上设置环境变量](#)”一节第 344 页

SALOGDIR 环境变量

指定 *backup.syb* 文件的位置。

语法

SALOGDIR=*directory-name*

注释

如果设置了 SALOGDIR 环境变量，则假定其中包含可写入备份历史记录文件 *backup.syb* 的目录路径。此文件将在每次执行 BACKUP 或 RESTORE 语句时更新。

在 Windows 上，*backup.syb* 文件创建于以下列表中的首个可写位置：

1. SALOGDIR 环境变量。
2. 安装目录。

在 32 位 Windows 平台上，缺省位置是 *install-dir\bin32*。如果此目录不存在，则给出一个错误。

3. 数据库服务器可执行文件的目录。
4. 将 *backup.syb* 文件写入当前驱动器的根目录。

在 Unix 上，*backup.syb* 文件创建于以下列表中的首个可写位置：

1. SALOGDIR 环境变量。
2. HOME 环境变量。
3. 将 *backup.syb* 文件写入数据库服务器启动时所在的目录。

另请参见

- “BACKUP 语句”一节 《SQL Anywhere 服务器 - SQL 参考》
- “在 Windows 上设置环境变量”一节第 344 页
- “在 Unix 和 Mac OS X 上设置环境变量”一节第 344 页

SATMP 环境变量

指定需要临时目录的数据库服务器和 SQL Anywhere 命令行实用程序所用的临时文件的位置。

语法

SATMP=*directory-name*

注释

SQL Anywhere 创建了两类临时文件：与数据库服务器有关的临时文件（在所有平台均创建）和与通信有关的临时文件（只在 Unix 上为客户端和服务端创建）。

SATMP 环境变量指定了需要临时目录的数据库服务器和 SQL Anywhere 命令行实用程序所用的临时文件的位置。它在数据库服务器作为服务运行时很有用，因为它使您能够将临时文件保存在其它程序无法访问的目录中。

如果在启动数据库服务器时未使用 `-dt` 选项指定临时文件的位置，则数据库服务器会检查 SATMP 环境变量的值以确定放置临时文件的位置。如果 SATMP 环境变量不存在，则使用现有的第一个 TMP、TMPDIR 或 TEMP 环境变量。在 Unix 上，如果以上环境变量均不存在，则使用 `/tmp`。

在 Windows Mobile 上，可在注册表中指定要用作服务器临时目录的目录。

有关 Windows Mobile 上临时文件位置的信息，请参见“[Windows Mobile 上的注册表设置](#)”一节第 374 页。

在 Unix 上，客户端和数据库服务器两者在通过共享内存连接时必须将 SATMP 设置为相同的值。

有关保护 Unix 上共享内存连接的信息，请参见“[安全提示](#)”一节第 1005 页。

如果想限制 Unix 数据库服务器或客户端所创建的临时文件的权限，必须将环境变量设置到不在以下列表中的目录：

- `/tmp`
- `/tmp/.SQLAnywhere`
- TMP 环境变量的值（如果已设置）
- TMPDIR 环境变量的值（如果已设置）
- TEMP 环境变量的值（如果已设置）
- 指向上面任何一个目录的符号链接

当 SATMP 设置到不在上述列表中的目录时，数据库服务器会搜索给定目录路径查找当前用户拥有的权限设置为 770、707 或 700 的目录。如果权限设置值不在这些值之中，就会创建权限设置值为 777 的文件。对每个找到的目录，数据服务器会从用于创建临时文件的权限掩码中删除相应的权限（分别是其它、组和其它 + 组）。

小心

将 SATMP 设置到不在上述列表中的目录中，可能会对用户以不同的 Unix 帐户通过共享内存连接到数据库服务器的造成障碍。

另请参见

- “-dt 服务器选项” 一节第 170 页
- “TMP、TEMPDIR 和 TEMP 环境变量” 一节第 366 页
- “在 Windows 上设置环境变量” 一节第 344 页
- “在 Unix 和 Mac OS X 上设置环境变量” 一节第 344 页
- “将不同的文件放置在不同的设备上” 一节 《SQL Anywhere 服务器 - SQL 的用法》

使用共享内存与旧版软件进行连接

在 SQL Anywhere 9 及更早的版本中，环境变量 ASTMP 等同于 SATMP。如果要使用共享内存来连接版本 9 和版本 10 的软件，必须设置 SATMP 和 ASTMP 环境变量以指定（相同的）临时目录的位置。

SHLIB_PATH 环境变量 [HP-UX]

在 HP-UX 上，该环境变量指定运行时在哪些目录中搜索 SQL Anywhere 应用程序所需的库。

语法

SHLIB_PATH=*path-list*

缺省值

- */opt/sqlanywhere11/lib32* (32 位平台)
- */opt/sqlanywhere11/lib64* (64 位平台)

注释

由安装程序创建的 *sa_config.sh* 和 *sa_config.csh* 文件是用于创建或修改上述及其它环境变量的脚本。

另请参见

- [“DYLD_LIBRARY_PATH 环境变量 \[Mac OS X\]” 一节第 346 页](#)
- [“LD_LIBRARY_PATH 环境变量 \[Linux 和 Solaris\]” 一节第 347 页](#)
- [“LIBPATH 环境变量 \[AIX\]” 一节第 348 页](#)
- [“在 Unix 和 Mac OS X 上设置环境变量” 一节第 344 页](#)

SQLANY11 环境变量

指定包含 SQL Anywhere 11 的目录的位置。

语法

SQLANY11=*directory-name*

缺省值

操作系统	位置
Windows	<i>C:\Program Files\SQL Anywhere 11</i>
AIX	<i>/usr/lpp/sqlanywhere11</i>
Mac OS X	<i>/Applications/SQLAnywhere11/System</i>
其它 Unix 操作系统	<i>/opt/sqlanywhere11</i>

注释

出于几个原因，应设置此环境变量。例如，示例需要使用此环境变量来查找 SQL Anywhere 应用程序。

在 Windows 上，安装程序会设置 SQLANY11 环境变量的位置。

在 Unix 上，由安装程序创建的 *sa_config.sh* 和 *sa_config.csh* 文件是用于创建或修改上述及其它环境变量的脚本。

另请参见

- “在 Windows 上设置环境变量” 一节第 344 页
- “在 Unix 和 Mac OS X 上设置环境变量” 一节第 344 页

SQLANYSAMP11 环境变量

指定 SQL Anywhere 示例目录的位置。

语法

SQLANYSAMP11=*directory-name*

缺省值

操作系统	缺省位置
Windows	<i>C:\Documents and Settings\All Users\Documents\SQL Anywhere 11\Samples</i>
Windows Vista	<i>C:\Users\Public\Documents\SQL Anywhere 11\Samples</i>
Mac OS X	<i>/Applications/SQLAnywhere11/Samples</i>
AIX	<i>/usr/lpp/sqlanywhere11/samples</i>
其它 Unix 操作系统	<i>/opt/sqlanywhere11/samples</i>

注释

在 Windows 上，安装程序会设置 SQLANYSAMP11 环境变量的位置。

在 Unix 上，由安装程序创建的 *sa_config.sh* 和 *sa_config.csh* 文件是用于创建或修改上述及其它环境变量的脚本。

另请参见

- “在 Windows 上设置环境变量” 一节第 344 页
- “在 Unix 和 Mac OS X 上设置环境变量” 一节第 344 页

SQLCONNECT 环境变量

在连接数据库服务器时指定其它连接参数。

语法

SQLCONNECT=*parameter=value*; ...

注释

此字符串是一个参数设置的列表，参数设置的形式为 *parameter=value*，由分号分隔。

如果在连接字符串中已经指定了连接参数，那么就不会使用 SQLCONNECT 环境变量指定的连接参数。

有关支持的连接参数的信息，请参见“[连接参数](#)”一节第 248 页。

口令安全风险

由于口令为明文形式，因此将其置入 SQLCONNECT 环境变量会带来安全风险。

另请参见

- [“解决连接参数冲突”一节第 81 页](#)
- [“在 Windows 上设置环境变量”一节第 344 页](#)
- [“在 Unix 和 Mac OS X 上设置环境变量”一节第 344 页](#)

SQLPATH 环境变量

指定命令文件和帮助文件的位置。

语法

SQLPATH=*path-list*

注释

Interactive SQL 在搜索系统路径之前，先在 SQLPATH 中所指定的目录内搜索命令文件和帮助文件。

另请参见

- [“在 Windows 上设置环境变量”一节第 344 页](#)
- [“在 Unix 和 Mac OS X 上设置环境变量”一节第 344 页](#)

SQLREMOTE 环境变量

该环境变量指定作为 SQL Remote FILE 消息链接的地址的子目录。

语法

SQLREMOTE=*path*

注释

SQL Remote 中的 FILE 消息链接的地址是 SQLREMOTE 环境变量的子目录。此环境变量应该指定一个共享目录。

在 Windows 操作系统（Windows Mobile 除外）上，设置 SQLREMOTE 环境变量的一种替代方法是将 *SQL Remote\Directory* 注册表条目设置为适当的根目录。

另请参见

- [“在 Windows 上设置环境变量”一节第 344 页](#)
- [“在 Unix 和 Mac OS X 上设置环境变量”一节第 344 页](#)

SYBASE 环境变量

该环境变量指定某些 Sybase 应用程序（包括 Adaptive Server Enterprise、Open Client、Open Server 和诸如 DSEdit 的实用程序）的安装主目录。

语法

SYBASE=*directory-name*

注释

只有在使用其它 Sybase 应用程序时，才需要设置此环境变量。

另请参见

- “在 Windows 上设置环境变量”一节第 344 页
- “在 Unix 和 Mac OS X 上设置环境变量”一节第 344 页

TMP、TMPDIR 和 TEMP 环境变量

指定 SQL Anywhere 临时文件的位置。

语法

TMP=*path*

TMPDIR=*path*

TEMP=*path*

注释

SQL Anywhere 软件可以为多种操作创建临时文件。临时文件在数据库服务器启动时创建，在数据库服务器停止时清除。顾名思义，临时文件是在数据库服务器运行时用来保存临时信息的。临时文件不保存需要在两个会话之间保留的信息。

临时文件存放在由 TMP、TMPDIR 或 TEMP 环境变量之一所指定的目录中。如果指定了这些环境变量中的多个，则使用 TMP、TMPDIR 和 TEMP 中的第一个。

SQL Anywhere 服务器首先检查 SATMP 环境变量。如果未指定该变量，则会检查这些环境变量。请参见“[SATMP 环境变量](#)”一节第 357 页。

如果所有这些环境变量都没有定义，则将临时文件放在服务器的当前工作目录中。如果未找到以上任何环境变量，则仅在 Unix 上才会使用 */tmp*。

在 Windows Mobile 上，可使用注册表指定要用作服务器临时目录的目录。

有关设置临时目录值的详细信息，请参见“[Windows Mobile 上的注册表设置](#)”一节第 374 页。

使用共享内存与旧版软件进行连接

在 SQL Anywhere 9 及更早的版本中，环境变量 ASTMP 等同于 SATMP。如果要使用共享内存来连接版本 9 和版本 10 的软件，必须设置 SATMP 和 ASTMP 环境变量来指定临时文件的位置。

另请参见

- “[-dt 服务器选项](#)”一节第 170 页
- “[SATMP 环境变量](#)”一节第 357 页
- “[在 Windows 上设置环境变量](#)”一节第 344 页
- “[在 Unix 和 Mac OS X 上设置环境变量](#)”一节第 344 页
- “[将不同的文件放置在不同的设备上](#)”一节 《[SQL Anywhere 服务器 - SQL 的用法](#)》

文件位置和安装设置

目录

安装目录结构	368
SQL Anywhere 如何定位文件	370
注册表和 INI 文件	373

安装目录结构

安装 SQL Anywhere 时会创建几个目录。这些目录中的某些文件是必需的，而其它文件不是必需的。本节介绍目录结构。

不管您所购买的 SQL Anywhere 软件是独立产品，还是捆绑在其它产品中，它都安装在一个单独的安装目录之下。SQLANY11 环境变量用于指定安装目录的位置。请参见“[SQLANY11 环境变量](#)”一节第 360 页。

SQL Anywhere 安装目录

SQL Anywhere 安装目录本身包含几个项目，其中包括：

- **自述文件** 名为 *readme.txt* 的自述文件包含最新的信息。

对于 Windows Mobile 以外的平台，在安装目录下有以下几个目录：

- **可执行文件目录** 每个操作系统平台都有一个单独的目录，其中包含配置文件和上下文相关帮助文件。

在 Windows 上（Windows Mobile 除外），这些文件安装在 *bin32* 或 *bin64* 目录中。如果使用的是 Unix，则它们安装在 *bin32* 或 *bin64* 及 *lib32* 或 *lib64* 目录中。

您只拥有操作系统版本所需的目录。

- **java 目录** JAR 文件存储在此目录中。
- **scripts 目录** scripts 目录包含数据库管理实用程序所使用的 SQL 脚本和用作示例的 SQL 脚本。
- **\SDK\Include 目录** \SDK\Include 目录中包含用于开发 SQL Anywhere 的 C/C++ 应用程序的头文件。在 Unix 上，此目录名为 *include*。

Windows Mobile 文件位置

在 Windows Mobile 设备上，所有文件都安装在安装目录 *\Program Files\SQLAny11* 中，只有 DLL 文件例外，这些文件安装在 *\Windows* 目录中。将不会创建子目录。

Unix 文件位置

语言资源安装在 *res* 目录中，共享对象安装在 *lib32* 或 *lib64* 目录中。

示例目录

安装 SQL Anywhere 11 时，可以选择示例的安装目录。文档中以 *samples-dir* 表示此位置。

SQLANYSAMP11 环境变量用于指定 *samples-dir* 的位置。请参见“[SQLANYSAMP11 环境变量](#)”一节第 361 页。

在 Windows 上，可以在 [开始] 菜单中顺次选择 [程序] » [SQL Anywhere 11] » [示例应用程序和项目] 来访问示例。

下表列出了 *samples-dir* 在所支持的每种操作系统中的缺省安装位置和典型安装位置：

操作系统	缺省安装位置 (<i>samples-dir</i>)	典型安装位置
Windows XP/200x	<i>%ALLUSERSPROFILE%\Documents\SQL Anywhere 11\Samples</i>	<i>C:\Documents and Settings\All Users\Documents\SQL Anywhere 11\Samples¹</i>
Windows Vista	<i>%PUBLIC%\Documents\SQL Anywhere 11\Samples</i>	<i>C:\Users\Public\Documents\SQL Anywhere 11\Samples</i>
Windows Mobile	<i>\Program Files\SQLAny11</i>	
Unix 和 Linux	<i>/opt/sqlanywhere11/samples</i>	提供了用于将示例复制到用户指定目录的脚本。
Mac OS X	<i>/Applications/SQLAnywhere11/Samples</i>	提供了用于将示例复制到用户指定目录的脚本。

¹ 在 Windows 资源管理器中访问 SQL Anywhere 示例目录时，其位置是 *Documents and Settings > All Users > 共享文档 > SQL Anywhere 11 > Samples*。不过，如果是通过命令提示符访问 SQL Anywhere 示例目录，则路径是 *C:\Documents and Settings\All Users\Documents\SQL Anywhere 11\Samples*。

SQL Anywhere 如何定位文件

Client Library 和数据库服务器为了以下两个主要目的需要定位文件：

- DLL 和初始化文件是运行 SQL Anywhere 所必需的。如果所定位的 DLL 不正确，则可能会出现版本不匹配错误。
- 有些文件在 SQL 语句（如 INSTALL JAVA 或 LOAD TABLE）中指定，因而需要在运行时进行定位。

使用文件名的 SQL 语句的示例包括以下语句：

- **INSTALL JAVA 语句** 指定包含 Java 类的文件的名称。
- **LOAD TABLE 和 UNLOAD TABLE 语句** 指定应从其装载数据或向其卸载数据的文件的名称。
- **CREATE DATABASE 语句** 此语句及可以创建文件的类似语句都需要有文件名。

在某些情况下，SQL Anywhere 使用简单的算法来定位文件。在其它情况下，则会执行更广泛的搜索。

简单文件搜索

在许多 SQL 语句（如 LOAD TABLE 或 CREATE DATABASE）中，文件名被解释为相对于数据库服务器的当前工作目录的文件名。

此外，当启动数据库服务器并提供数据库文件名（DatabaseFile (DBF) 参数）时，路径被解释为相对于当前工作目录的路径。

在 Windows 上进行广泛的文件搜索

在 Windows 上，SQL Anywhere 程序（包括数据库服务器和管理实用程序）可执行更广泛的搜索来查找所需的文件，如 DLL 或共享库。在上述情况下，SQL Anywhere 程序按以下顺序查找文件：

1. 模块的目录（程序可执行文件或库文件所在的目录）。
2. 可执行文件目录（程序可执行文件或库所在的目录）。
3. 安装路径（SQL Anywhere 安装目录、*install-dir*）。*install-dir* 是由 SQLANY11 环境变量所指定的单一目录（如果已定义）。
4. 无路径（当前工作目录）。
5. Location 注册表条目。
6. 系统特定的目录。此类目录包括存放公用操作系统文件的目录，如 Windows 操作系统上的 *Windows* 目录和 *Windows\system32* 目录。
7. PATH 目录。搜索系统路径和用户路径中的目录?&

注意

在 Windows 上，SQL Anywhere 搜索以下相对于前面列表中各个位置的路径：

1. .
2. ..
3. *.\bin32* 和 *..\bin32*（仅适用于 32 位程序）
4. *.\bin64* 和 *..\bin64*（仅适用于 64 位程序）
5. *.\java*（用于与 Java 相关的文件）
6. *..\java*（用于与 Java 相关的文件）
7. *.\scripts*（用于 SQL 脚本文件）
8. *..\scripts*（用于 SQL 脚本文件）

在 Windows Mobile 上进行广泛的文件搜索

在 Windows Mobile 上，SQL Anywhere 程序（包括数据库服务器和管理实用程序）可执行更广泛的搜索来查找所需的文件，如 DLL 或共享库。在上述情况下，SQL Anywhere 程序按以下顺序查找文件：

1. 模块的目录（程序可执行文件或库文件所在的目录）。
2. 可执行文件目录（程序可执行文件或库所在的目录）。
3. 无路径（当前工作目录）。
4. Location 注册表条目。
5. 系统特定的目录?&此类目录包括存放公用操作系统文件的目录，如 *Windows*。

注意

在 Windows Mobile 上，SQL Anywhere 搜索以下相对于前面列表中各个位置的路径：

1. .
2. ..
3. *.\bin32*
4. *..\bin32*
5. *.\java*（用于与 Java 相关的文件）
6. *..\java*（用于与 Java 相关的文件）
7. *.\scripts*（用于 SQL 脚本文件）
8. *..\scripts*（用于 SQL 脚本文件）

在 Unix 上进行广泛的文件搜索

在 Unix 上，SQL Anywhere 程序（包括数据库服务器和管理实用程序）可执行更广泛的搜索来查找所需的文件，如 DLL 或共享库。在上述情况下，SQL Anywhere 程序按以下顺序查找文件：

1. 可执行文件路径（如果可以确定）。
2. 安装路径（SQL Anywhere 安装目录、*install-dir*）。*install-dir* 是由 SQLANY11 环境变量所指定的单一目录（如果已定义）。
3. 无路径（当前工作目录）。
4. PATH 环境变量。
5. LIBPATH 环境变量：
 - Linux 和 Solaris 上为 LD_LIBRARY_PATH
 - HP-UX 上为 LD_LIBRARY_PATH 和 SHLIB_PATH
 - AIX 上为 LIBPATH
 - Mac OS X 上为 DYLD_LIBRARY_PATH

注意

在 Unix 上，SQL Anywhere 搜索以下相对于前面列表中各个位置的路径：

1. .
2. ..
3. *./bin32* 和 *../bin32*（仅适用于 32 位程序）
4. *./bin64* 和 *../bin64*（仅适用于 64 位程序）
5. *./lib32* 和 *../lib32*（仅适用于 32 位程序的库文件）
6. *./lib64* 和 *../lib64*（仅适用于 64 位程序的库文件）
7. *./java*（用于与 Java 相关的文件）
8. *../java*（用于与 Java 相关的文件）
9. *./scripts*（用于 SQL 脚本文件）
10. *../scripts*（用于 SQL 脚本文件）
11. *./res*（用于 *.res* 文件）
12. *../res*（用于 *.res* 文件）
13. *./tix*（用于 *.tix* 文件）
14. *../tix*（用于 *.tix* 文件）

注册表和 INI 文件

在除 Windows Mobile 以外的 Windows 操作系统上，SQL Anywhere 使用多个注册表设置。在 Unix 上，这些设置则保存在初始化文件中。

软件安装时会自动进行这些设置，执行常规操作时一般不需要访问注册表或初始化文件。此处提供这些设置是为了方便那些需要对其操作环境进行修改的人员。

可以使用文件隐藏实用程序，通过简单加密技术对 SQL Anywhere 所使用的 .ini 文件的内容进行模糊处理。请参见“[文件隐藏实用程序 \(dbfhide\)](#)”一节第 724 页。

小心

在 Unix 上，除非仅使用 SQL Anywhere 数据源，否则不应使用文件隐藏实用程序 (dbfhide) 对系统信息文件（缺省名称是 `.odbc.ini`）进行简单加密。如果计划使用其它数据源（例如，为了 MobiLink 同步），那么对系统信息文件内容进行模糊处理可能会妨碍其它驱动程序正常运行。

当前用户和本地计算机设置

某些操作系统具有两个级别的系统设置。某些设置特定于单个用户且仅当该用户登录时才使用；这些设置称为**当前用户**设置。某些设置是应用于整个计算机的全局设置，所有用户都会应用这些设置；这些设置称为**本地计算机**设置。只有计算机上拥有管理员权限的用户才能更改本地计算机设置。

SQL Anywhere 对当前用户设置和本地计算机设置都给予支持。例如，在 Windows XP 上，这些设置分别保存在 `HKEY_CURRENT_USER` 和 `HKEY_LOCAL_MACHINE` 注册表项中。

当前用户优先

如果在当前用户注册表和本地计算机注册表中都进行了某个设置，则当前用户设置优先于本地计算机设置。

何时需要本地计算机设置

如果将 SQL Anywhere 程序作为**服务**来运行，则应确保在**本地计算机**级别进行这些设置。

注销计算机时只要不是将计算机完全关闭，这些服务就会使用特殊帐户继续运行。可以使服务不依赖于单个帐户，因此需要访问本地计算机设置。

除 SQL Anywhere 程序外，某些 Web 服务器也作为服务运行。必须进行本地计算机设置，以便 Apache 或 IIS 可以与这样的 Web 服务器配合工作。

一般情况下，建议使用本地计算机设置。

注册表结构

在 Windows（Windows Mobile 除外）上，可以通过注册表编辑器直接访问注册表。SQL Anywhere 注册表条目保存在 `HKEY_CURRENT_USER` 或 `HKEY_LOCAL_MACHINE` 注册表项中的以下位置：

```
Software
  Sybase
    SQL Anywhere
      11.0
    Sybase Central
      6.0.0
```

修改注册表存在危险

修改注册表的风险由用户自行承担。建议修改注册表前先备份系统。

安装时的注册表设置

在 Windows 上，安装程序会在 `HKEY_LOCAL_MACHINE\Software\Sybase` 注册表中进行以下设置。以下列表描述了这些注册表设置中的一部分：

- **SQL Anywhere\11.0\Location** 此条目保存 SQL Anywhere 软件的安装目录位置。例如：

```
Location "c:\Program Files\SQL Anywhere 11"
```

- **SQL Anywhere\11.0\Samples Location** 此条目保存示例程序的安装目录位置。例如：

```
Samples Location "C:\Documents and Settings\All Users\Documents\SQL Anywhere 11\Samples\"
```

- **SQL Anywhere\11.0\Online Resources** 此条目保存 [在线资源] 文档的位置。例如：

```
Online Resources "c:\Program Files\SQL Anywhere 11\support\ianywhere.html"
```

- **SQL Anywhere\11.0\Language** 此条目保存一个由两个字母组成的代码，用来表示消息和错误的当前语言。例如：

```
Language "EN"
```

根据安装时指定的语言选择来设置语言。请参见“[了解区域设置语言](#)”一节第 389 页。

- **Sybase Central\6.0.0\Language** 此条目保存一个由两个字母组成的代码，用来表示消息和错误的当前语言。例如：

```
Language "EN"
```

此条目由 Sybase Central 使用。根据安装时指定的语言选择设置语言。请参见“[了解区域设置语言](#)”一节第 389 页。

Windows Mobile 上的注册表设置

可以通过在注册表中设置以下值来指定在 Windows Mobile 上想要将哪个目录用作服务器的临时目录：

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

`TempFolder` 是想要使用的临时目录的名称。该服务器执行以下操作之一：

- 使用指定的目录（如果该目录存在）。
- 创建指定的目录（如果该目录不存在），只要父目录已经存在。

如果指定的目录不存在并且无法创建，数据库服务器将：

- 使用 *\Temp* 目录（如果该目录存在）。
- 尝试创建 *\Temp* 目录（如果该目录不存在）。

如果 *\Temp* 目录不存在且无法创建，服务器将使用当前目录。

国际语言和字符集

目录

SQL Anywhere 的本地化版本	378
了解字符集	384
了解区域设置	389
了解归类	392
有关国际语言和字符集的任务	399
字符集和归类参考信息	404

SQL Anywhere 的本地化版本

本地化是指一种产品在语言和文化方面与目标地区相适应，它通常是语言和国家/地区的组合。本地化会影响许多组件，其中包括打包、安装、文档、软件用户界面以及错误消息/警告消息/信息性消息。

SQL Anywhere 软件已被本地化为五种语言：

- 英语
- 法语
- 德语
- 日语
- 简体中文

语言选择在安装时予以确定。

该文档的本地化版本包括英语、德语、日语和简体中文。

在 Windows 上，利用 **[开始]** 菜单项可以在已安装语言与英语之间对软件进行重新配置。利用语言选择实用程序 (dblang) 可以将软件重新配置为任何可用的语言，包括附加部署语言。请参见“[Windows 上的部署软件本地化](#)”一节第 379 页和“[语言选择实用程序 \(dblang\)](#)”一节第 745 页。

下表按操作系统平台显示了每种语言的可用性。

平台	英语	法语	德语	日语	简体中文
Windows	是	是	是	是	是
Windows Mobile	是	是	是	是	是
Linux	是		是	是	是
Unix	是				
Mac OS X	是				

软件和文档完全本地化

SQL Anywhere for Windows 有以下几种语言版本，适用于开发、部署和管理：

- 英语
- 法语
- 德语
- 日语
- 简体中文

对于英语、德语、日语和简体中文，所有 SQL Anywhere 组件都已本地化，包括：

- 打包
- 安装程序
- 文档和上下文相关帮助
- 软件
 - [开始] 菜单项和程序文件夹
 - 数据库服务器和客户端库
 - MobiLink 服务器和客户端
 - SQL Remote 客户端
 - 管理工具，包括 Interactive SQL 和 Sybase Central 以及所有相关插件
 - 命令行工具，例如 dbinit 和 dbunload

对于法语，安装程序、软件和上下文相关帮助已本地化。

以下组件未进行本地化，仅有英语版本：

- DataWindow .NET
- InfoMaker
- PowerDesigner Physical Data Model

Windows 上的部署软件本地化

除了前面列出的五种主要语言外，SQL Anywhere 还提供了以下语言的部署软件资源：

- 意大利语
- 朝鲜语
- 立陶宛语
- 波兰语
- 葡萄牙语（巴西）
- 俄语
- 西班牙语
- 繁体中文
- 乌克兰语

部署本地化适用于通常向最终用户部署的软件组件子集。打包、文档、管理、部署和安装软件未进行本地化。已本地化的软件组件包括：

- 数据库服务器和客户端库
- MobiLink 服务器和客户端
- SQL Remote 客户端
- 命令行工具，例如 dbinit 和 dbunload

SQL Anywhere 的国际功能

国际化是指软件能够独立于软件运行的语言或操作系统来处理各种语言及其相应的字符集。SQL Anywhere 具有完全的国际化功能。本文档从以下几个特性讨论了最常被要求和使用的功能。

- **Unicode 支持** SQL Anywhere 对 Unicode 提供如下支持：
 - 在用于 ODBC、OLE DB、ADO.NET 和 JDBC 的 SQL Anywhere 客户端库中对 UTF-16 的客户端支持
 - 用于以 UTF-8 存储 Unicode 字符数据的 NCHAR 数据类型
 - CHAR 数据类型可以使用 UTF-8 编码
- **代码页和字符集** SQL Anywhere 数据库服务器及相关工具支持 Windows (ANSI/ISO)、UTF-8 及 Unix 代码页和字符集。
- **归类** SQL Anywhere 支持两种归类算法：SQL Anywhere 归类算法 (SACA) 以及使用 Unicode 的国际组件 (ICU) 的 Unicode 归类算法 (UCA)。

有关 ICU 的详细信息，请参见“[何谓 ICU、何时需要它？](#)”一节第 381 页。

SACA 以语言正确性为代价提供了快速、紧凑而又合理的排序。UCA 具备语言正确性，但在存储需求和执行时间方面要付出小小的代价。请参见“[了解归类](#)”一节第 392 页。

对于高级排序和比较功能，SQL Anywhere 还提供了 SORTKEY 和 COMPARE 函数。这些函数提供了高级语言排序功能，类似于字典或电话簿中使用的排序。在合适的地方，还提供了不区分大小写和不区分重音的排序和比较。请参见“[SORTKEY 函数 \[String\]](#)”一节《[SQL Anywhere 服务器 - SQL 参考](#)》和“[COMPARE 函数 \[String\]](#)”一节《[SQL Anywhere 服务器 - SQL 参考](#)》。

SQL Anywhere 还包含特意为在字符列上自动使用基于 SORTKEY 的排序而设计的功能。当对字符列使用 ORDER BY 时，`sort_collation` 数据库选项会指定要使用的排序顺序。还可以使用计算列来存储字符列的排序键，这样便无需在每次指定 ORDER BY 时都对其进行计算了。请参见“[sort_collation 选项 \[数据库\]](#)”一节第 544 页。

- **字符集转换** SQL Anywhere 会在服务器和客户端系统上的字符集编码之间转换数据，并保持数据的完整性，即使在混合字符集环境中也是如此。请参见“[字符集转换](#)”一节第 386 页。
- **标识符** SQL Anywhere 支持使用包含大多数单字节和多字节字符的标识符，而无需使用引号。例外情况通常只限于空格和标点符号。
- **货币** 支持对货币符号（包括欧元符号）进行排序。SQL Anywhere 不提供货币格式设置支持。
- **日期和时间格式** SQL Anywhere 支持公历，并且提供了各种日期和时间字符串格式。可使用 `date_format`、`time_format` 和 `timestamp_format` 数据库选项进行自定义格式设置。`date_format` 和 `timestamp_format` 选项在缺省情况下为日期的 ISO 兼容格式，YYYY-MM-DD。SQL Anywhere 提供了 CONVERT 函数，利用它可将日期和时间的输出格式转换成各种常用的格式。请参见：
 - “[date_format 选项 \[数据库\]](#)”一节第 496 页
 - “[time_format 选项 \[兼容性\]](#)”一节第 553 页
 - “[timestamp_format 选项 \[兼容性\]](#)”一节第 554 页
 - “[CONVERT 函数 \[Data type conversion\]](#)”一节《[SQL Anywhere 服务器 - SQL 参考](#)》

另请参见

- “使用指定归类创建数据库”一节第 401 页
- “建议的字符集和归类”一节第 408 页

何谓 ICU、何时需要它？

ICU（即 Unicode 的国际组件）是一个由 IBM 开发并维护的开放源代码库。ICU 通过提供 Unicode 支持促进了软件国际化。SQL Anywhere 使用 ICU 实现了某些字符集转换和归类操作。

数据库服务器上何时需要 ICU？（除 Windows Mobile 之外的所有平台）

理想情况下，ICU 应始终可供数据库服务器使用。下表指出了何时以及因何需要 ICU：

需要 ICU 的情况……	注意
UCA 用作 NCHAR 或 CHAR 字符集的归类。	UCA 需要 ICU。
数据库字符集不是 UTF-8，而是多字节字符集。	用于将口令从数据库字符集转换为 UTF-8（数据库口令在内部以 UTF-8 存储）。
客户端字符集与数据库字符集不同，而且其中之一是多字节字符集（包括 UTF-8）。这包括 Unicode ODBC、OLE DB、ADO.NET 和 iAnywhere JDBC 应用程序，不管数据库字符集为何，至少有一个客户端没有 ICU。	与多字节字符集的来回转换需要 ICU。
数据库字符集不是 UTF-8，且需要在 CHAR 和 NCHAR 值之间进行转换。	数据库服务器需要使用 ICU 将 UTF-8 转换为另一字符集。
嵌入式 SQL 客户端使用了 UTF-8 以外的 NCHAR 字符集。	数据库服务器需要使用 ICU 将 UTF-8 转换为另一字符集。请注意，缺省的嵌入式 SQL 客户端 NCHAR 字符集与初始客户端 CHAR 字符集相同。可使用 <code>db_change_nchar_charse</code> 函数对此进行更改。请参见“ db_change_nchar_charset 函数 ”一节《SQL Anywhere 服务器 - 编程》。
使用了 CSCONVERT 或 SORTKEY 函数。调用 CSCONVERT 函数是为了在符合上述第三项要求的字符集之间进行转换。	在上述第三项要求的情况下进行转换字符集时需要 ICU。许多 sortkey 标签的 Sortkey 生成都需要 UCA，进而需要 ICU。请参见“ CSCONVERT 函数 [String] ”一节《SQL Anywhere 服务器 - SQL 参考》和“ SORTKEY 函数 [String] ”一节《SQL Anywhere 服务器 - SQL 参考》。

数据库服务器上何时需要 ICU? (Windows Mobile)

下表指出了 Windows Mobile 何时以及因何需要 ICU:

需要 ICU 的情况……	注释
UCA 用作 NCHAR 归类或 CHAR 归类。	UCA 需要 ICU。
使用了 SORTKEY 函数。	许多 sortkey 标签的 Sortkey 生成都需要 UCA, 进而需要 ICU。请参见“ SORTKEY 函数 [String] ”一节《 SQL Anywhere 服务器 - SQL 参考 》。
CHAR 字符集与 OS 字符集不匹配。	即使这些字符集匹配, 如果使用的是 NCHAR, 或 CHAR 字符集为多字节, 也建议使用 ICU 来增强字符集转换。

注意

如果不安装 ICU 库, 则在创建数据库时, 必须选择其字符集与 Windows Mobile 字符集匹配的归类或 UTF8BIN 归类作为 CHAR 归类。同样, 在创建数据库时, 必须选择 UTF8BIN 归类作为 NCHAR 归类。

何时在数据库服务器上不使用 ICU 就能获得正确的字符集转换?

在以下两种情况下, 不使用 ICU 就能获得正确的字符集转换: 数据库字符集和客户端字符集均为单字节且 *sqlany.cvf* 可用 (所有平台); 操作系统支持该转换 (仅限 Windows)。这是因为只要 *sqlany.cvf* 文件可用或主机操作系统安装了相应的转换器, 就可以不使用 ICU 而进行单字节到单字节的转换。

客户端上何时需要 ICU? (除 Windows Mobile 之外的所有平台)

对于 Unicode 客户端应用程序来说, 无论数据库字符集为何, 当所有客户端都安装了 ICU 时, 均有可能提高客户端和数据库服务器的组合性能。这是因为可以将某些必需的转换活动从数据库服务器下放到客户端, 并且所需进行的转换更少。

此外, 如果是在 Windows 平台上使用 ODBC, 则必须在客户端安装 ICU, 甚至对 ANSI 应用程序也是如此。这是因为驱动程序管理器会将 ANSI ODBC 调用转换为 Unicode ODBC 调用。

字符集问题及解答

下表指明可在何处查找问题的答案。

要回答此问题……	请考虑阅读……
如何确定为数据库使用哪个归类?	“了解归类”一节第 392 页
在软件中, 尤其是在 SQL Anywhere 中, 字符是如何表示的?	“了解字符集”一节第 384 页

要回答此问题……	请考虑阅读……
SQL Anywhere 提供了哪些归类？	“选择归类”一节第 395 页
SQL Anywhere 支持哪些字符集编码？	“支持的字符集”一节第 404 页
我的客户端计算机上的字符集与数据库中正在使用的字符集不同。如何在客户端和服务端之间正确地交换字符？	“字符集转换”一节第 386 页
哪些字符集可用于连接字符串？	“连接字符串和字符集”一节第 386 页
如何更改现有数据库的归类序列？	“将数据库从一种归类更改为另一种归类”一节第 402 页

了解字符集

本节提供了与国际语言和字符集有关的软件问题的一般信息。

字符集、编码和归类概述

每个软件都会使用一个**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。

为在内部正确地表示这些字符，每个软件都会运用某种**编码**，又称作**字符编码**。通过编码这种方法可将每个字符均映射到一个或多个字节的信息，并将其表示为十六进制数。编码的一个例子是 UTF-8。

有时可以交替使用术语字符集和编码，因为这两个方面密切相关。

代码页是编码的一种形式。代码页是字符到数字表示的映射，数字通常介于整数 0 到 255 之间。代码页的一个例子就是 Windows 代码页 1252。

就本文档而言，术语编码、字符编码、字符集编码以及代码页都是同一个意思。

数据库服务器为了对字符进行排序（例如，按字母顺序列出名称）使用了**归类**。归类是字符编码（字符与其表示之间的映射）和字符**排序顺序**的组合。对于每个字符集都可能多个排序顺序；例如，区分大小写的顺序和不区分大小写的顺序，或者两种语言可以按不同顺序对相同字符进行排序。

字体用于打印或在屏幕上显示字符，它是字符集中的字符与其外观之间的映射。字体由操作系统来处理。

操作系统还使用**键盘映射**将键盘上的按键或按键组合映射到字符集中的字符。

客户端/服务器计算中的语言问题

使用客户端应用程序的数据库用户可以从以下来源查看或访问字符串：

- **数据库中的数据** 字符串和其它文本数据存储在数据库中。在对请求进行响应时，数据库服务器处理这些字符串。例如，可以要求数据库服务器提供表中以排在 N 之前的字母开头的的所有姓氏。这就需要执行字符串比较，并采取某种字符集排序。
- **数据库服务器软件消息** 应用程序可能导致产生数据库错误。例如，应用程序可能提交一个引用不存在的列的查询。在这种情况下，数据库服务器会返回一条警告或错误消息。此消息存放在**语言资源库**中，该库是 SQL Anywhere 所使用的一个 DLL 或共享库。
- **客户端应用程序** 客户端应用程序界面显示文本，并且客户端应用程序可能会在内部处理文本。
- **客户端软件消息** 客户端库与数据库服务器使用同一语言库，以便为客户端应用程序提供消息。
- **操作系统** 客户端及服务器操作系统可以提供消息或处理文本。

为了获得令人满意的工作环境，所有这些文本源必须一起工作。一般说来，它们都必须使用用户的语言和/或字符集进行工作。

单字节字符集

许多语言只有很少的字符，用单字节字符集表示就足够了。在此类字符集中，每个字符都以单个字节来表示：两位十六进制数。

用单个字节最多可以表示 256 个字符。没有一个单字节字符集可以包含国际上使用的所有字符，其中包括带重音符号的字符。通过开发一组代码页解决了这一问题，每个代码页代表适合于一种或多种国家/地区语言的一个字符集。例如，代码页 1253 包含希腊语字符集，代码页 1252 包含西欧语言。代码页有很多，代码页的名称也有很多。以上示例为 Windows 的代码页。

上部页和下部页

除了少数例外情况，对于所有的代码页，字符 0 到 127 都是相同的。此字符范围的映射称为 **ASCII** 字符集。它包括大写和小写英语字母以及常见的标点符号和阿拉伯数字。此范围通常称为**七位范围**（因为只需七位即可代表数字 0 到 127）或**下部页**。从 128 到 255 的字符称为**扩展字符**或**上部代码页字符**，它们随代码页的不同而不同。

如果仅使用英语字母表中的字符，则很少会出现代码页兼容性问题，因为这些字符都是在各代码页的 ASCII 部分（0 到 127）表示的。但是，如果使用了其它字符（这种情况在任何非英语环境中都很常见），则当数据库和应用程序使用不同的代码页时可能会出现问題。

例如，假定一个使用 UTF-8 字符集的数据库从包含 cp1252 数据的文件中装载了一个表，并且未在 LOAD TABLE 语句中将编码指定为 cp1252。由于未指定编码，数据将被假定采用 UTF-8 编码，因而不会进行任何字符转换；cp1252 编码直接存储在数据库中。这意味着不会将诸如欧元符号（在 cp1252 中表示为十六进制 80）等字符转换成 UTF-8。采用 UTF-8 的欧元符号是以三字节序列 E2 82 AC 表示的，但是，在此情况下，它将以 80 形式存储在数据库中。此后，当应用程序请求数据时，数据库服务器会尝试将数据从 UTF-8 转换为客户端字符集。该转换会产生乱码。

多字节字符集

某些语言（如日语和中文）的字符数远远超过 256 个。这些字符无法使用单字节来表示，因此必须使用多字节编码对其进行编码。另外，某些字符集为了包含更大数量的字符，以便在一个更全面的字符集中表示来自多种语言的字符，也使用了多字节形式。其中的一个例子就是 UTF-8。

多字节字符集可以具有**可变宽度**，由此，某些字符是单字节字符，其它则是双字节等字符。

有关多字节字符集和归类的详细信息，请参见“[SQL Anywhere 归类算法 \(SACA\)](#)”一节第 392 页。

示例

举例来说，代码页 932（日语）中字符的长度或者是一个字节或者是两个字节。如果首字节的值（又称为**前导字节**）在 \x81 到 \x9F 或 \xE0 到 \xFC（十进制值 129 到 159 或 224 到 252）的十六进制值范围内，则该字符是一个双字节字符，而随后的那个字节（又称为**后随字节**）与首字节共同构成了该字符。后随字节是除首字节之外的一个或多个字节。

如果首字节在前导字节范围之外，则该字符是一个单字节字符，而下一个字节是下一个字符的首字节。

Windows 中的 ANSI 和 OEM 代码页

对于 Windows 用户而言，总共会使用两个代码页。使用 Windows 图形用户界面的应用程序使用 Windows 代码页。Windows 代码页与 ISO 字符集兼容，也与 ANSI 字符集兼容。它们通常称为 **ANSI 代码页**。

Windows 中的字符模式的应用程序（使用命令提示符窗口的应用程序）使用过去在 DOS 中使用的代码页。由于历史原因，这些代码页称为 **OEM（原始设备制造商）代码页**。

SQL Anywhere 既支持基于 OEM 代码页的归类，又支持基于 ANSI 代码页的归类。OEM 归类是为了保持兼容性而提供的，不应将其用于新数据库。请参见“[支持的归类 and 替代归类](#)”一节第 404 页。

SQL Anywhere 数据库中的字符集

SQL Anywhere 数据库可以使用一个或两个字符集（编码）来存储字符数据。CHAR 数据类型（包括 CHAR、VARCHAR 和 LONG VARCHAR）使用单字节或多字节字符集。可以使用 UTF-8。NCHAR 数据类型（包括 NCHAR、NVARCHAR 和 LONG NVARCHAR）使用 UTF-8。

使用 LOAD TABLE 语句以及诸如 CSCONVERT、TO_CHAR 和 TO_NCHAR 函数时，可将数据库字符集称为 db_charset，将数据库 NCHAR 字符集称为 nchar_charset。

有关 CHAR 和 NCHAR 数据类型的详细信息，请参见“[字符数据类型](#)”一节《[SQL Anywhere 服务器 - SQL 参考](#)》。

字符集转换

SQL Anywhere 可以在代表相同字符但这些字符在字符集或代码页中处于不同位置的字符集之间执行字符集转换。两个字符集间必须存在某种程度的兼容性，这样才能进行转换。例如，在 EUC-JIS 与 cp932 字符集之间能够进行字符集转换，但在 EUC-JIS 与 cp1252 之间则不能进行转换。

SQL Anywhere 使用 Unicode 的国际组件 (ICU) 开放源代码库（由 IBM 开发并维护）来实现字符集转换。

有关为比较不同数据类型的值而进行字符集转换的详细信息，请参见“[数据类型之间的比较](#)”一节《[SQL Anywhere 服务器 - SQL 参考](#)》。

连接字符串和字符集

如果并非所有的客户端都使用相同的字符集，则连接字符串在字符集转换期间可能是一个难题。这是因为连接字符串会由客户端库进行分析，以查找或启动数据库服务器。但是，此分析是在不知道数据库服务器所使用的字符集或语言的情况下进行的。

接口库通过执行如下操作来分析连接字符串：

1. 将连接字符串分解成它的 *keyword=value* 对。只要您不使用大括号 {} 将 CommLinks (LINKS) 连接参数括起来，就可以独立于字符集执行此分解操作。请使用推荐的圆括号 () 将参数括起来。在某些多字节字符集中，大括号是有效的**后随字节**（除首字节之外的字节）。
2. 查找服务器。不会对服务器名进行任何字符集转换。如果客户端字符集和服务器字符集不同，则在服务器名中使用扩展字符可能导致找不到服务器。
如果客户端和服务器分别运行不同的操作系统或区域设置，则应在服务器名中使用 7 位 ASCII 字符。
3. 将 DatabaseName (DBN) 或 DatabaseFile (DBF) 连接参数从客户端字符集转换为数据库服务器字符集。
4. 找到数据库后，便会将其余的连接参数转换为数据库的字符集。

SQL 语句和字符集

SQL Anywhere 服务器字符集转换，会在分析和执行之前将所有 SQL 语句转换为数据库字符集。此转换的副作用是，任何不能在 SQL 语句中转换为数据库字符集的字符会转换为替换字符。通过以下方法之一可以执行包含任意一个 Unicode 字符的 SQL 语句：

- 使用 UNISTR 函数指定 Unicode 字符值
- 使用主机变量指定 Unicode 字符值
- 将 UTF-8 用作数据库字符集

如果选择 UTF8BIN 作为 char 归类，则数据库字符集为 UTF-8。如果指定 UTF-8 编码，则 char 归类是 UCA。

Unicode 归类算法 (UCA) 提供高级的比较、排序和大小写转换，但会影响性能。虽然 UTF8BIN 既节省空间又快，但其排序顺序和比较是二进制的。如果在 SQL 语句中需要 Unicode 字符但不需要 UCA 的全部功能来进行排序和比较，则可将 char 归类指定为 UTF8BIN。仅在必要时通过 SORTKEY 和 COMPARE 函数来使用 UCA。

另请参见

- “[SORTKEY 函数 \[String\]](#)” 一节 《[SQL Anywhere 服务器 - SQL 参考](#)》
- “[COMPARE 函数 \[String\]](#)” 一节 《[SQL Anywhere 服务器 - SQL 参考](#)》
- “[Unicode 归类算法 \(UCA\)](#)” 一节第 393 页
- “[SQL Anywhere 归类算法 \(SACA\)](#)” 一节第 392 页

查看数据时排除意外符号问题

使用 Interactive SQL 等客户端应用程序来选择并查看数据时，数据中的字符可能出现意外符号，如方块、箭头及问号。

发生这种情况主要有两个原因。第一个原因是存储在数据库中的基础数据存在问题。例如，如果在向数据库中插入数据时需要进行字符集转换，而原始字符集中的某些字符在数据库字符集中不存在等同的字符，则会插入替换字符来代替。

客户端应用程序中出现意外字符的第二个原因，也是更为常见的原因是，用于显示数据的字体不支持某些字符。将字体更改为 Unicode 字体可以解决此问题。如果不能为客户端应用程序更改字体，也可以更改操作系统的缺省字体。

例如，假定您使用 Windows 系统，该系统使用的标准英文字体 (Tahoma) 不支持日语字符的显示。但是，数据库字符集为 cp932 且数据库中包含日语数据，在查询数据库时，结果中的日语字符会显示为小框。在 Interactive SQL 中，您可以更改用于显示结果的字体，方法是选择 [工具] » [选项] » [结果] » [字体]，然后指定某个 Unicode 字体，如 Arial Unicode MS 或 Lucida Sans Unicode。Unicode 字体是理想的选择，因为这类字体可以显示许多语言的字符。

如果所使用的客户端应用程序未提供用户可更改的字体设置，则它可能使用操作系统的缺省字体。这种情况下，请参考该操作系统文档中有关如何更改系统的缺省字体的信息，然后将系统的缺省字体更改为某种 Unicode 字体。

另请参见

- “替换字符”一节 《SQL Anywhere 服务器 - SQL 参考》
- “使用 Interactive SQL”一节第 636 页

区分大小写的国际角度考虑

对于诸如系统视图名和列名之类的标识符，SQL Anywhere 始终保留大小写而不区分大小写。名称以其创建时的大小写进行存储，但是对标识符的任何访问都是以不区分大小写的方式进行的。

例如，系统视图的名称以大写形式 (SYSDOMAIN、SYSTAB 等) 存储，但是对它们的访问是不区分大小写的，因此以下两条语句是等同的：

```
SELECT * FROM systab;  
SELECT * FROM SYSTAB;
```

大小写字符的等同性是在归类中定义的。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。例如，土耳其语归类的大小写转换行为会导致意外而又微妙的错误。最常见的错误是找不到包含字母 **I** 或 **i** 的系统对象。

有关土耳其语字符集和归类的详细信息，请参见“土耳其语字符集和归类”一节第 411 页。

了解区域设置

数据库服务器和客户端库都使用**区域设置定义**来识别其语言和字符集环境。

区域设置简介

当客户端或客户端库向数据库服务器发出请求时，它会使用应用程序区域设置（亦即客户端区域设置）来确定返回结果应采用的字符集以及错误消息、警告和其它消息的语言。数据库服务器会将自身的区域设置与应用程序区域设置进行比较，以确定是否需要进行字符集转换。服务器上的不同数据库可能具有不同的区域设置定义，并且每个客户端都可能具有各自的区域设置。

区域设置包含以下组成部分：

- **语言** 语言以一个使用 ISO-639 标准值的双字符字符串来表示（例如，德语为 DE）。数据库服务器和客户端都有代表各自区域设置的语言值。

数据库服务器使用区域设置语言来确定要装载的语言库。创建数据库时，如果没有指定归类，则数据库服务器还会结合使用语言和字符集来确定要使用的归类。

客户端库使用区域设置语言来确定要装载的语言库以及要从数据库请求的语言。请参见“[了解区域设置语言](#)”一节第 389 页。

- **字符集** 字符集是使用中的代码页（亦即编码）。客户端和服务端都有字符集值，而且它们的字符集值可能不同。如果不同，则会使用字符集转换来实现互操作性。请参见“[了解区域设置字符集](#)”一节第 391 页。

了解区域设置语言

区域设置语言是客户端应用程序的用户正在使用的语言，或者是数据库服务器的用户应当使用的语言。有关如何查找区域设置的详细信息，请参见“[确定区域设置信息](#)”一节第 399 页。

客户端库和数据库服务器都以相同方式确定区域设置的语言：

1. 使用 SALANG 环境变量的值（如果存在）。请参见“[SALANG 环境变量](#)”一节第 355 页。
2. 对于 Windows 而言，如果 SALANG 环境变量不存在，请检查 SQL Anywhere 语言注册表条目。请参见“[安装时的注册表设置](#)”一节第 374 页。
3. 检查操作系统语言设置。
4. 如果通过以上设置仍无法确定语言，则缺省情况下为英语。

语言标签值

下表显示了有效的语言标签值以及等同的 ISO 639 语言代码：

语言	ISO_639 语言代码	语言标签	替代标签
阿拉伯语	AR	arabic	N/A

语言	ISO_639 语言代码	语言标签	替代标签
捷克语	CS	czech	N/A
丹麦语	DA	danish	N/A
荷兰语	NL	dutch	N/A
英语	EN	us_english	english
芬兰语	FI	finnish	N/A
法语	FR	french	N/A
德语	DE	german	N/A
希腊语	EL	greek	N/A
希伯来语	HE	hebrew	N/A
匈牙利语	HU	hungarian	N/A
意大利语	IT	italian	N/A
日语	JA	japanese	N/A
朝鲜语	KO	korean	N/A
立陶宛语	LT	lithuanian	N/A
挪威语	NO	norwegian	norweg
波兰语	PL	polish	N/A
葡萄牙语	PT	portuguese	portugue
俄语	RU	russian	N/A
简体中文	ZH	chinese	simpchin
西班牙语	ES	spanish	N/A
瑞典语	SV	swedish	N/A
泰语	TH	thai	N/A
繁体中文	TW	tchinese	tradchin
土耳其语	TR	turkish	N/A

语言	ISO_639 语言代码	语言标签	替代标签
乌克兰语	UK	ukrainian	N/A

了解区域设置字符集

应用程序区域设置定义和服务器区域设置定义都包含字符集。从数据库服务器请求字符串时，应用程序会使用它的字符集。数据库服务器会将数据库字符集与应用程序的字符集进行比较，以确定是否需要进行字符集转换。如果数据库服务器无法与客户端字符集进行来回转换，则连接将会失败。

1. 如果设置了 SACHARSET 环境变量，则使用它的值来确定字符集。请参见“[SACHARSET 环境变量](#)”一节第 352 页。

仅在创建新数据库且在当时没有指定归类时，数据库服务器才会使用 SACHARSET。

2. 如果连接字符串指定了字符集，则使用该字符集。有关详细信息，请参见“[CharSet 连接参数 \[CS\]](#)”一节第 252 页。
3. Open Client 应用程序会检查 Sybase 版本目录的 *locales* 子目录中的 *locales.dat* 文件。
4. 使用操作系统的字符集信息确定区域设置：
 - 在 Windows 操作系统上，使用的是当前 Windows ANSI 代码页。
 - 在 Unix 平台上，将以指定顺序检查以下地区环境变量：LC_ALL、LC_MESSAGES、LC_CTYPE、LANG。若找到的其中第一个环境变量已设置，则使用它的值来确定字符集。如果无法从操作系统确定字符集，则使用 iso_1（也称为 Windows 代码页 28591、ISO 8859-1 Latin I、ISO 8859-1 Latin-1 或 iso_8859-1:1987）的缺省值。
5. 在其它任何平台上，均使用缺省代码页 cp1252。

有关如何查找区域设置的详细信息，请参见“[确定区域设置信息](#)”一节第 399 页。

了解归类

归类描述了如何对来自特定字符集或编码的字符进行排序和比较。SQL Anywhere 支持两种归类算法：SQL Anywhere 归类算法 (SACA) 和 Unicode 归类算法 (UCA)。SACA 以语言正确性为代价提供了快速、紧凑而又合理的排序。UCA 具备语言正确性，但在存储需求和执行时间方面要付出小小的代价。

本节介绍了提供的归类，并提供了有关使用哪些归类的建议。

有关如何使用特定归类创建数据库的详细信息，请参见“[使用指定归类创建数据库](#)”一节第 401 页和“[初始化实用程序 \(dbinit\)](#)”一节第 729 页。

有关使用归类定制语法来自定义 UCA 归类的信息，请参见“[归类定制选项](#)”一节第 396 页。

SQL Anywhere 归类算法 (SACA)

SQL Anywhere 归类算法提供了合理的比较、排序以及单字节和多字节字符集的大小写转换。该算法既节省空间又快。字符串的映射形式（如索引）与原始字符串长度相同。用于比较、排序和大小写转换的映射使用了对字符串每字节值的简单表查找。

SACA 自从 SQL Anywhere 的早期形式 Watcom SQL 开始就提供了。

单字节字符集

在单字节字符集的典型归类中，字符的所有重音和非重音形式都被映射到相同的值，不区分归类重音。相同字母的重音和非重音形式在比较上完全相等且彼此一起进行排序。

归类还提供了大小写字母之间的转换，保留重音。

多字节字符集

在多字节字符集中，前导字节被映射成 256 个不同的值。后随字节使用其二进制值进行比较。

对于多字节字符集的大多数归类而言，这种映射技术提供了合理的排序，因为字符集编码将字符分组由前导字节标识的 256 字节页。这些页和每页内的字符在字符集中都按合理的顺序排列。归类通常会在字符集内保留页（前导字节）的排序。某些页可能按其它特征排序。例如，为日语代码页 932 提供的 932JPN 归类用于将全宽 (Kanji) 和半宽 (katakana) 字符分组。

仅为 7 位英语字符提供大小写转换。

UTF-8 字符集

UTF-8 是一个多字节字符集。每个字符都包含一个到四个字节。SQL Anywhere 提供了用于对 UTF-8 字符进行排序的 UTF8BIN 归类。

在 UTF8BIN 中，前导字节被映射成 256 个不同的值，后随字节使用其二进制值进行比较。由于 UTF-8 中字符的表示形式以及 256 不同映射值的限制，无法对相关字符（如同一字母的重音和非重音形式）进行分组。排序实质是二进制排序。

仅对 7 位英语字符支持大小写转换。

Unicode 归类算法 (UCA)

Unicode 归类算法是用于对整个 Unicode 字符集进行排序的一种算法。它提供语言方面的正确比较、排序和大小写转换。UCA 是作为 Unicode 标准的一部分进行开发的。SQL Anywhere 使用 Unicode 的国际组件 (ICU) 开放源代码库（由 IBM 开发并维护）来实现 UCA。

注意

采用缺省的 UCA 排序可按适当的顺序对大多数语言中的大多数字符进行排序。但是，由于在共享字符的各种语言之间，在排序和比较方面存在诸多变化，UCA 无法为所有语言提供正确的排序。为此，ICU 提供了用于定制 UCA 的语法。请参见“[归类定制选项](#)”一节第 396 页。

UCA 以空间和时间上的小小代价提供了高级的比较、排序和大小写转换。

字符串的映射形式比原始字符串要长。通过该算法可对更加复杂的字符进行精密处理。

与 SQL Anywhere 归类算法不同，Unicode 归类算法仅用于单字节和 UTF-8 字符集，它将每个字符分成一个或多个属性。对于字母而言，这些属性为基本字符、重音和大小写。

非字母通常只有一个属性，基本字符。

UCA 按如下方式进行字符串比较：

- 比较基本字符。如果一个基本字符字符串与另一个不同，则比较完成。不考虑重音和大小写。
- 如果数据库区分重音，则比较重音。如果重音不同，则比较完成。不考虑大小写。
- 如果数据库区分大小写，则比较每个字符的大小写。

当且仅当两个字符串的基本字符、重音和大小写都相同时，原始字符串值才相等。

示例

假定使用 UCA 比较下表第一列中的字符串。后续各列描述了每个字符串的三个属性。请注意，基本字符完全相同，单词仅在重音和大小写上不同。

字符串	基本字符	重音	大小写
noel	noel	无、无、无、无	小写、小写、小写、小写
noël	noel	无、无、重音、无	小写、小写、小写、小写
Noel	noel	无、无、无、无	大写、小写、小写、小写
Noël	noel	无、无、重音、无	大写、小写、小写、小写

下表显示了使用 UCA 时在四种可能的区分重音和区分大小写的组合中将会进行的排序：

区分重音	区分大小写	ORDER BY 结果	解释
N	N	Noel、noël、Noël 和 noel, 顺序随意	<ul style="list-style-type: none"> ● 忽略重音 ● 忽略大小写 ● 认为所有值都相等 ● 在四元组内随机排序
Y	N	Noel 和 noel, 顺序随意 其后是 noël 和 Noël, 顺序随意	<ul style="list-style-type: none"> ● 非重音在重音之前, 因此 e 在 ë 前面 ● 忽略大小写, N 和 n 在每个两元组中都以随机顺序排列
N	Y	Noel 和 Noël, 顺序随意 其后是 noël 和 noel, 顺序随意	<ul style="list-style-type: none"> ● 大写字母在小写字母之前, 因此 N 在 n 前面 ● 忽略重音, e 和 ë 在每个两元组中都以随机顺序排列
Y	Y	Noel noel Noël noël	<ul style="list-style-type: none"> ● 非重音在重音之前, 因此 e 在 ë 前面 ● 大写字母在小写字母之前, 因此 N 在 n 前面

SQL Anywhere 数据库中的归类

CHAR 归类

CHAR 数据类型（包括 CHAR、VARCHAR 和 LONG VARCHAR）可以使用采取 SQL Anywhere 归类算法的归类，也可以使用 Unicode 归类算法。无论是哪种情况，所使用的归类均称为 CHAR 归类。

NCHAR 归类

NCHAR 数据类型（包括 NCHAR、NVARCHAR 和 LONG NVARCHAR）可以使用 Unicode 归类算法，也可以使用 UTF8BIN 归类（它使用 SQL Anywhere 归类算法）。

选择区分大小写和区分重音

创建 SQL Anywhere 数据库时，如果未指定区分大小写，则该数据库将不区分大小写。通过指定相应的选项，可以使其区分大小写。在已经创建了数据库之后，只有重建数据库才能更改区分大小写设置。

数据库区分大小写决定了 SACA 归类和 UCA 归类都区分大小写，因而也决定了 CHAR 归类和 NCHAR 归类都区分大小写。

创建 SQL Anywhere 数据库时，如果未指定区分重音，则该数据库将不区分重音。通过指定相应的选项，可以使其区分重音。在已经创建了数据库之后，只有重建数据库才能更改区分重音设置。

数据库区分重音仅会影响 UCA 归类，无论它用于 CHAR 归类、NCHAR 归类还是两者，都是如此。如果为 CHAR 归类和 NCHAR 归类均选择了 SACA 归类，则区分重音选项不起作用。区分重音是 SACA 归类的一个属性，不能在创建数据库时使用所提供的选项来指定。

选择归类

创建数据库时，SQL Anywhere 可以根据操作系统的语言和字符集设置选择缺省归类。在大多数情况下，所选的缺省归类是合适的，但是您也可以从所提供的各种归类中显式地选择一种归类，以满足您的需要。在某些情况下，SQL Anywhere 对某种特定语言支持多种归类。

所选择的归类应该使用适合于数据库中数据的字符集和排序顺序。您还可以指定归类定制选项，以便能够对字符的排序和比较进行更多的控制。有关创建数据库的详细信息，请参见“[创建数据库](#)”一节第 20 页。

有关数据排序和国际功能的详细信息，请参见“[SQL Anywhere 的国际功能](#)”一节第 380 页。

另请参见

- “[使用指定归类创建数据库](#)”一节第 401 页
- “[建议的字符集和归类](#)”一节第 408 页
- “[CREATE DATABASE 语句](#)”一节 《[SQL Anywhere 服务器 - SQL 参考](#)》
- “[初始化实用程序 \(dbinit\)](#)”一节第 729 页

选择归类时的注意事项

在为数据库选择归类时，请考虑以下因素：

- 在使用字符集转换时，除了性能开销之外，还会额外增加系统配置的复杂性。所选择的归类要避免进行字符集转换。如果数据库服务器和客户端使用同一字符集，则不会使用字符集转换。
通过在数据库中使用与客户端计算机操作系统上正在使用的字符集相匹配的归类序列，可以避免字符集转换。如果在客户端计算机上使用的是 Windows 操作系统，请选择 ANSI 字符集。
- 如果客户端计算机使用多种字符集，或数据库必须存储 Unicode 数据，请考虑使用 UCA 和/或 UTF8BIN 归类。但是，请注意，UCA 归类不能与除 UTF-8 之外的多字节字符集一起使用。
- 所选择的归类要使用适合于数据库中数据的字符集和排序顺序。通常会有几种归类都满足这一要求。

归类定制选项

如果在创建数据库时选择 UCA 归类，则可以选择指定归类定制选项。如果未选择 UCA 归类，仍然可以使用定制语法来控制是否区分大小写。也可以在对数据进行比较或排序时，使用 COMPARE 和 SORTKEY 函数指定定制选项。

归类定制选项以 "关键字-值" 对的形式表示。以下是受支持的关键字的表，包括关键字的允许的替代形式和允许的值。

注意
使用归类定制选项创建的数据库无法使用 10.0.1 之前版本的数据库服务器启动。

关键字	归类	替代形式	允许值
Locale	UCA	(无)	任何有效区域设置代码。例如，en。
CaseSensitivity	支持的全部归类	CaseSensitive、Case	<ul style="list-style-type: none"> ● respect 区分字母间的大小写差异。对于 UCA 归类，这等效于 UpperFirst。对于其它归类，它取决于归类本身。 ● ignore 忽略字母间的大小写差异。 ● UpperFirst 始终将大写排在前面 (Aa)。 ● LowerFirst 始终将小写排在前面 (aA)。
AccentSensitivity	UCA	AccentSensitive、Accent	<ul style="list-style-type: none"> ● respect 区分字母间的重音差异。 ● ignore 忽略字母间的重音差异。 ● 法语 使用法语规则区分重音的不同。

关键字	归类	替代形式	允许值
PunctuationSensitivity	UCA	PunctuationSensitive、Punct	<ul style="list-style-type: none"> ● ignore 忽略标点的差异。 ● primary 使用一级排序（只考虑字母）。例如，a > b。 ● quaternary 使用四级排序：首先考虑字母，然后依次是大小写、重音和标点。例如，multiByte、multibyte、multi-byte 和 multi-Byte 的排序是： <ul style="list-style-type: none"> ○ multiByte ○ multibyte ○ multi-Byte ○ multi-byte <p>您无法对不区分大小写或重音的数据库指定四级排序。</p>
SortType	UCA	(无)	<p>要使用的排序类型。可能的值：</p> <ul style="list-style-type: none"> ● phonebook ● traditional ● standard ● pinyin ● stroke ● direct ● posix ● big5han ● gb2312han <p>有关这些排序类型的详细信息，请参见 Unicode Technical Standard #35，其网址是 http://www.unicode.org/reports/tr35/。</p>

注意

要定制符合瑞典学院 2005 标准的 UCA 归类（根据该标准，在主级别中将 V 和 W 视为不同的字符），请指定 UCA (locale=swe;sorttype=phonebook)。如果不指定 sorttype=phonebook，则在瑞典语区域设置中会将 V 和 W 视为相同的字符。

另请参见

- “CREATE DATABASE 语句”一节 《SQL Anywhere 服务器 - SQL 参考》
- “初始化实用程序 (dbinit)”一节第 729 页
- “COMPARE 函数 [String]”一节 《SQL Anywhere 服务器 - SQL 参考》
- “SORTKEY 函数 [String]”一节 《SQL Anywhere 服务器 - SQL 参考》

SQL Anywhere 如何为新数据库选择缺省归类

创建新数据库时，如果没有显式指定归类，则 SQL Anywhere 将使用语言和字符集来确定归类。

- 语言来自于 SALANG 环境变量（如果存在）、注册表或操作系统。请参见“SALANG 环境变量”一节第 355 页。
- 字符集来自于 SACHARSET 环境变量（如果存在）或操作系统。请参见“SACHARSET 环境变量”一节第 352 页。

有关国际语言和字符集的任务

本节汇总了与国际语言和字符集问题有关的各种任务。

确定缺省归类

如果在创建数据库时没有显式指定归类，则会使用缺省归类。缺省归类取决于您正在使用的操作系统。

◆ 确定计算机的缺省归类

1. 启动 Interactive SQL。
2. 连接到示例数据库。
3. 输入以下查询：

```
SELECT PROPERTY( 'DefaultCollation' );
```

即会返回缺省归类。

有关此归类的详细信息，请参见“选择归类”一节第 395 页。

确定区域设置信息

可以使用诸如 PROPERTY、DB_PROPERTY 和 CONNECTION_PROPERTY 等函数来确定区域设置信息。下表显示了如何使用这些函数返回有关客户端连接、数据库和数据库服务器的区域设置信息。

系统函数和参数	返回值
<code>SELECT PROPERTY('CharSet');</code>	数据库服务器的字符集。通常是服务器所在计算机的字符集。
<code>SELECT PROPERTY('DefaultCollation');</code>	数据库服务器用来创建数据库的缺省 CHAR 归类。
<code>SELECT PROPERTY('DefaultNcharCollation');</code>	数据库服务器用来创建数据库的缺省 NCHAR 归类。
<code>SELECT PROPERTY('Language');</code>	数据库服务器的区域设置语言。
<code>SELECT DB_PROPERTY('CharSet');</code>	用于在数据库中存储 CHAR 数据的字符集。
<code>SELECT DB_PROPERTY('NcharCharSet');</code>	用于在数据库中存储 NCHAR 数据的字符集。

系统函数和参数	返回值
<code>SELECT DB_PROPERTY('MultiByteCharSet');</code>	CHAR 数据是否使用多字节字符集（On=是，Off=否）。
<code>SELECT DB_PROPERTY('Language');</code>	以逗号分隔的双字母代码列表，这些代码表示数据库 CHAR 归类所支持的语言。
<code>SELECT DB_PROPERTY('Collation');</code>	数据库服务器所使用的 CHAR 归类名称。
<code>SELECT DB_PROPERTY('NcharCollation');</code>	数据库服务器所使用的 NCHAR 归类名称。
<code>SELECT CONNECTION_PROPERTY('CharSet');</code>	用于连接的 CHAR 数据字符集。
<code>SELECT CONNECTION_PROPERTY('NcharCharSet');</code>	用于连接的 NCHAR 数据的字符集。
<code>SELECT CONNECTION_PROPERTY('Language');</code>	用于连接的客户端语言。

另请参见

- “PROPERTY 函数 [System]” 一节 《SQL Anywhere 服务器 - SQL 参考》
- “DB_PROPERTY 函数 [System]” 一节 《SQL Anywhere 服务器 - SQL 参考》
- “CONNECTION_PROPERTY 函数 [System]” 一节 《SQL Anywhere 服务器 - SQL 参考》

设置区域设置

可以使用操作系统上的缺省区域设置，也可以显式设置供计算机上的 SQL Anywhere 组件使用的区域设置。

◆ 设置 SQL Anywhere 区域设置

1. 如果缺省区域设置符合您的需要，则无需采取任何操作。

有关如何查明操作系统的缺省区域设置的详细信息，请参见“确定区域设置信息”一节第 399 页。

2. 如果需要更改区域设置，可以同时设置 SALANG 和 SACHARSET 环境变量，也可以设置其中的一个：

```
SACHARSET=charset
SALANG=language-code
```

charset 是有效的字符集标签，*language_code* 是来自有效语言列表的语言代码。请参见“语言标签值”一节第 389 页。

有关在不同的操作系统上设置环境变量的详细信息，请参见“SQL Anywhere 环境变量”第 343 页。

使用指定归类创建数据库

在创建数据库时，可以为每个数据库指定归类。根据数据库服务器计算机的操作系统的代码页和语言，可以推断出缺省归类。

有关使用 NCHAR 归类的信息，请参见“[NCHAR 归类](#)”一节第 394 页。

◆ 在创建数据库时指定数据库归类（命令提示符）：

1. 运行以下命令以列出推荐的归类序列：

```
dbinit -l
```

列表的第一列是归类标签，它是您在创建数据库时要提供的。

2. 使用 `dbinit` 实用程序创建数据库，使用 `-z` 选项指定归类序列。以下命令使用希腊语归类创建了一个数据库。

```
dbinit -z 1253ELL mydb.db
```

以下命令创建区分大小写的数据库，`spanish.db`，它对非 NCHAR 数据使用 1262spa 归类。对于 NCHAR 数据，将指定 UCA 归类，区域设置为 `es`，并且按小写字母在前排序。

```
dbinit -c -z 1252spa -zn uca(locale=es;case=LowerFirst) spanish.db
```

◆ 在创建数据库时指定数据库归类 (SQL)：

- 可以使用 `CREATE DATABASE` 语句创建数据库。以下语句使用希腊语归类创建了一个数据库：

```
CREATE DATABASE 'mydb.db'  
COLLATION '1253ELL';
```

以下语句使用代码页 1252 创建数据库，并将 UCA 用于 CHAR 和 NCHAR 数据类型。在比较和排序期间考虑重音和大小写。

```
CREATE DATABASE 'c:\\uca.db'  
COLLATION 'UCA'  
ENCODING 'CP1252'  
NCHAR COLLATION 'UCA'  
ACCENT RESPECT  
CASE RESPECT;
```

◆ 在创建数据库时指定数据库归类 (Sybase Central)：

- 可以使用 Sybase Central 中的 [\[创建数据库向导\]](#) 来创建数据库。在该向导的一个页面中，可从列表中选择一种归类。



另请参见

- “教程：创建 SQL Anywhere 数据库” 一节 《SQL Anywhere 服务器 - SQL 的用法》
- “创建数据库” 一节第 20 页
- “CREATE DATABASE 语句” 一节 《SQL Anywhere 服务器 - SQL 参考》
- “初始化实用程序 (dbinit)” 一节第 729 页

将数据库从一种归类更改为另一种归类

将数据库更改为另一种归类需要重建数据库。归类是在数据库创建时选择的，且无法进行更改。

◆ 更改归类

1. 按如下方式确定现有数据库的字符集：

```
SELECT DB_PROPERTY( 'CharSet' );
```

对于 SQL Anywhere 的早期版本，此属性可能不存在。归类名称也暗示了字符集。例如，归类 1252LATIN1 使用代码页 1252。

2. 确定现有数据库中数据的字符集。

此字符集应该与数据库字符集相同或与其兼容。若非如此，就有充分的理由重建数据库，但是在重建过程中需要非常小心。

特别是，如果您一直都在将采用 850LATIN1 归类的数据库与 SQL Anywhere 的早期版本一起使用，而这些版本不支持字符集转换（版本 5 和更早版本）或在缺省情况下禁用字符集转换（版本 6 和 7），并且您的客户端应用程序是普通 Windows 应用程序，则在您的数据库中可能会存在代码页 1252 字符数据，而该数据库期望的数据应采用代码页 850。对此情况的简单测试就是使用带有 ENCODING 选项的 UNLOAD TABLE 卸载某些字符数据，然后用 Windows 记事本查看它。如果重音数据正确，则数据库中的字符数据与 Windows ANSI 代码页匹配，对于英语和其它西欧语言，该代码页为 1252。如果数据在基于 DOS 的编辑器中显示正确，则字符数据与 Windows OEM 代码页匹配，该代码页很可能是 437 或 850。

3. 卸载数据库。

如果数据字符集与数据库字符集不兼容，在不进行字符集转换的情况下卸载数据就显得至关重要。根据正在使用的 SQL Anywhere 版本，可以使用 dbunload 内部卸载功能，也可以使用 UNLOAD TABLE 语句手工卸载数据。

4. 创建新数据库，指定要使用的归类和字符集。

5. 将数据装载到新的数据库中。

如果卸载的数据和模式 (*reload.sql*) 与用来进行重装的计算机字符集匹配，则可以使用 dbunload 的外部重装选项。服务器的字符集转换会自动将数据转换为正确的数据库字符集。

如果数据的编码与数据库的字符集不匹配，并且装载数据时使用的是 LOAD TABLE 语句（内部重装），则必须使用 ENCODING 子句；在缺省情况下，数据库服务器不会对使用 LOAD TABLE 语句装载的数据执行字符集转换。

如果数据的编码与正在使用的计算机的代码页不匹配，并且装载时使用的是 INPUT 语句（外部重装），则必须使用 ENCODING 子句；否则，数据库服务器会假定数据采用计算机的本地字符集。

另请参见

- “LOAD TABLE 语句”一节 《SQL Anywhere 服务器 - SQL 参考》
- “UNLOAD 语句”一节 《SQL Anywhere 服务器 - SQL 参考》
- “INPUT 语句 [Interactive SQL]”一节 《SQL Anywhere 服务器 - SQL 参考》
- “使用指定归类创建数据库”一节第 401 页
- “卸载实用程序 (dbunload)”一节第 795 页
- “重建数据库”一节 《SQL Anywhere 服务器 - SQL 的用法》
- “CharSet 连接参数 [CS]”一节第 252 页

字符集和归类参考信息

以下各节提供了有关 SQL Anywhere 的字符集和归类的信息。

支持的字符集

SQL Anywhere 支持几百个字符集和标签，该名单还在不断增长。大量名称或标签都能识别字符集编码。要查看 SQL Anywhere 所支持的字符集列表，请运行以下命令：

```
dbinit -le
```

每行输出均针对给定的字符集编码以逗号分隔形式列出了最常见的标签。每行输出中的第一个标签是字符集编码的首选 SQL Anywhere 名称。其余则是不同机构、组织或标准所使用的标签。它们是 IANA（Internet 编号授权委员会）、MIME（多用途 Internet 邮件扩展）、ICU（Unicode 的国际组件）、Java 和 ASE (Adaptive Server Enterprise)。

如果您没有找到所要查找的字符集，还可以执行以下命令来查看包含不常见标签的更长列表：

```
dbinit -le+
```

指定字符集编码标签后，SQL Anywhere 将在其能够识别的标签集中搜索该标签。有时，不同的机构会将相同的标签用于不同的字符集。SQL Anywhere 会尽全力根据上下文解析多义性。例如，如果 JDBC 应用程序通过多义性标签引用字符集，则会将该标签解析为 Java 标准标签。建议始终使用 SQL Anywhere 标签以避免任何多义性。[International Components for Unicode](#) 是了解有关字符集编码标签的最佳资源。

除了通过 dbinit -le 选项返回的字符集编码标签外，还可以使用以下字符集别名：

- **os_charset** 作为数据库服务器的主机的操作系统所使用的字符集的别名。
- **char_charset** 数据库使用的 CHAR 字符集的别名。
- **nchar_charset** 数据库使用的 NCHAR 字符集的别名。

确定是否支持某个字符集或标签的简便方法是使用 C\$CONVERT 函数对它进行测试。请参见“[C\\$CONVERT 函数 \[String\]](#)”一节《[SQL Anywhere 服务器 - SQL 参考](#)》。

支持的归类和替代归类

下表列出了 SQL Anywhere 版本 8 中引入的可与 SORTKEY 和 COMPARE 函数一起使用的兼容性归类。

在 SQL Anywhere 版本 10 中，SORTKEY 和 COMPARE 函数的实现方式已更改为 ICU（International Components for Unicode，Unicode 的国际组件）和 UCA（Unicode Collation Algorithm，Unicode 归类算法），且归类名现在映射到 UCA 归类。SQL Anywhere 版本 10 及更高版本中的排序和比较，可能与 SQL Anywhere 的先前版本不同。

另请参见

- “COMPARE 函数 [String]” 一节 《SQL Anywhere 服务器 - SQL 参考》
- “SORTKEY 函数 [String]” 一节 《SQL Anywhere 服务器 - SQL 参考》
- “何谓 ICU、何时需要它？” 一节第 381 页

归类标签	说明
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 2312-80 8 位编码
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
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, 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 拉丁语 5, 土耳其语, 带有扩展的 ISO 8859-9
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, 立陶宛语
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

归类标签	说明
EUC_KOREA	朝鲜语 KS C 5601-1992 编码, Johab
EUC_TAIWAN	繁体中文 Big 5 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, 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
UCA	标准缺省 UCA 归类
UTF8BIN	UTF-8, Unicode 的 8 位多字节编码, 二进制排序

替代归类

替代归类是为了与旧版本的 SQL Anywhere 保持兼容性而提供的, 也可用于特殊用途。要查看所支持替代归类的完整列表, 请运行以下命令:

```
dbinit -l+
```

Adaptive Server Enterprise 归类

在下表中列出了所支持的 Adaptive Server Enterprise 归类, 这些归类和 SORTKEY 函数之类的功能一起使用。

说明	归类名	归类 ID
缺省 Unicode 多语言	default	0
CP 850 备选: 无重音	altnoacc	39
CP 850 备选: 小写字母优先	altdict	45
CP 850 西欧: 不分大小写, 优先	altnocsp	46
CP 850 斯堪的纳维亚文字典	scandict	47
CP 850 斯堪的纳维亚: 不分大小写, 优先	scannocp	48
GB 拼音	gbpinyin	无
二进制排序	binary	50
拉丁文-1 英语、法语、德语字典	dict	51

说明	归类名	归类 ID
拉丁文-1 英语、法语、德语，不分大小写	nocase	52
拉丁文-1 英语、法语、德语，不分大小写，优先	nocasep	53
拉丁文-1 英语、法语、德语，无重音	noaccent	54
拉丁文-1 西班牙语字典	espdict	55
拉丁文-1 西班牙语，不分大小写	espnoacs	56
拉丁文-1 西班牙语，无重音	espnoaac	57
ISO 8859-5 俄语字典	rusdict	58
ISO 8859-5 俄语，不分大小写	rusnoacs	59
ISO 8859-5 西里尔字母字典	cyrdict	63
ISO 8859-5 西里尔字母，不分大小写	cyrnoacs	64
ISO 8859-7 希腊文字典	elldict	65
ISO 8859-2 匈牙利文字典	hundict	69
ISO 8859-2 匈牙利文，无重音	hunnoaac	70
ISO 8859-2 匈牙利文，不分大小写	hunnoacs	71
ISO 8859-5 土耳其文字典	turdict	72
ISO 8859-5 土耳其文，无重音	turnoaac	73
ISO 8859-5 土耳其文，不分大小写	turnoacs	74
CP 874 (TIS 620) 皇家泰语字典	thaidict	1
ISO 14651 排序标准	14651	22
Shift-JIS 二进制顺序	sjisbin	179
Unicode UTF-8 二进制排序	utf8bin	24
EUC JIS 二进制顺序	eucjisbn	192
GB2312 二进制顺序	gb2312bn	137
CP932 MS 二进制顺序	cp932bin	129

说明	归类名	归类 ID
Big5 二进制顺序	big5bin	194
EUC KSC 二进制顺序	euckscbn	161

建议的字符集和归类

尽管 SQL Anywhere 识别几百个字符集、代码页、编码以及归类的名称，但本节根据所使用的语言提供了建议与 Windows 和 Unix 平台一起使用的那些名单。

可以使用 `dbinit -le` 选项来获取 SQL Anywhere 数据库的所有可用字符集编码列表。请参见“[初始化实用程序 \(dbinit\)](#)”一节第 729 页。

注意

对于在下表中找不到的语言，建议选择使用 UTF-8 编码的 UCA 或 UTF8BIN 归类。

Windows 平台

语言	Windows 代码页	字符集标签	归类	替代归类
阿拉伯语	1256	Windows-1256	1256ARA	
中欧和东欧语言	1250	Windows-1250	1250LATIN2	
丹麦语	1252	Windows-1252	1252LATIN1	
荷兰语	1252	Windows-1252	1252LATIN1	
英语	1252	Windows-1252	1252LATIN1	
芬兰语	1252	Windows-1252	1252SWEFIN	
法语	1252	Windows-1252	1252LATIN1	
德语	1252	Windows-1252	1252LATIN1	
希腊语	1253	Windows-1253	1253ELL	
希伯来语	1253	Windows-1253	1255HEB	
意大利语	1252	Windows-1252	1252LATIN1	
日语	932	Windows-31J	932JPN	
朝鲜语	949	IBM949	949KOR	

语言	Windows 代码页	字符集标签	归类	替代归类
立陶宛语	1257	Windows-1257	1257LIT	
挪威语	1252	Windows-1252	1252NOR	
波兰语	1250	Windows-1250	1250POL	
葡萄牙语	1252	Windows-1252	1252LATIN1	
俄语	1251	Windows-1251	1251CYR	
简体中文	936	GBK	936ZHO	
西班牙语	1252	Windows-1252	1252SPA	
瑞典语	1252	Windows-1252	1252SWEFIN	
泰语	874	TIS-620	874THAIBIN	
繁体中文 - 中国香港特别行政区	950	Big5-HKSCS	950ZHO_HK	
繁体中文 - 中国台湾地区	950	Big5	950ZHO_TW	
土耳其语	1254	Windows-1254	1254TRK	1254TRKALT
乌克兰语	1251	Windows-1251	1251CYR	
西欧语言	1252	Windows-1252	1252LATIN1	

Unix 平台

语言	字符集标签	Collation	替代归类
阿拉伯语	ISO_8859-6:1987	UCA	
中欧和东欧语言	ISO_8859-2:1987	UCA	
丹麦语	ISO-8859-15	ISO9LATIN1	ISO1LATIN1
荷兰语	ISO-8859-15	ISO9LATIN1	ISO1LATIN1
英语	ISO-8859-15	ISO9LATIN1	ISO1LATIN1
芬兰语	ISO-8859-15	ISO9LATIN1	ISO1LATIN1
法语	ISO-8859-15	ISO9LATIN1	ISO1LATIN1

语言	字符集标签	Collation	替代归类
德语	ISO-8859-15	ISO9LATIN1	ISO1LATIN1
希腊语	ISO_8859-7:1987	UCA	
希伯来语	ISO_8859-8:1988	UCA	
意大利语	ISO-8859-15	ISO9LATIN1	ISO1LATIN1
日语	EUC-JP ¹	EUC_JAPAN	
朝鲜语	EUC-KR	EUC_KOREA	
立陶宛语	(使用 UTF-8)	UCA 或 UTF8BIN	
挪威语	ISO-8859-15	ISO9LATIN1	ISO1LATIN1
波兰语	ISO_8859-2:1987	UCA	
葡萄牙语	ISO-8859-15	ISO9LATIN1	ISO1LATIN1
俄语	ISO_8859-5:1988	UCA	
简体中文	GB2312	EUC_CHINA	
西班牙语	ISO-8859-15	ISO9LATIN1	ISO1LATIN1
瑞典语	ISO-8859-15	ISO9LATIN1	ISO1LATIN1
泰语	(使用 UTF-8)	UCA 或 UTF8BIN	
繁体中文 - 中国香港特别行政区	Big5-HKSCS	950ZHO_HK	950TWN
繁体中文 - 中国台湾地区	EUC-TW	EUC_TAIWAN	
繁体中文 - 中国台湾地区	Big5	950ZHO_TW	
土耳其语	ISO_8859-9:1989	920TRK	
乌克兰语	ISO_8859-5:1988	UCA	
西欧语言	ISO-8859-15	ISO9LATIN1	ISO1LATIN1

¹ EUC-JP 是 SQL Anywhere 标签 Extended_UNIX_Code_Packed_Format_for_Japanese 的替代标签。

土耳其语字符集和归类

土耳其语中有两种看起来像字母 **I** 的形式。一种形式称为 **I-dot**，如下所示：

i, İ

第二种形式称为 **I-no-dot**，如下所示：

ı, I

虽然这些字母看似同一字母的变体，但在土耳其语字母表中，它们被视为截然不同的字母。为了支持这些变体，SQL Anywhere 提供了土耳其语归类 1254TRK。

土耳其语中这些字符的大小写转换规则与 ANSI SQL 中的大小写转换标准规则不兼容。例如，在土耳其语中与 **ı** 等同的小写形式为：

ı

但是，在 ANSI 标准中则为：

i

因此，不区分大小写的匹配正确与否取决于要匹配的文本是土耳其语还是英语/ANSI。在许多环境下，并没有足够的信息来进行这样的区分，这就导致此类数据库中的一些非标准行为。

例如，请考虑针对使用 1254TRK 归类的数据库执行的以下语句：

```
SELECT * FROM syshistory    //actual table name is SYSHISTORY
SELECT * FROM fig          //actual name is FIG
```

第一条语句引用了一个系统对象，并且需要用 ANSI SQL 转换规则来匹配该名称。第二条语句引用了一个用户对象，并且需要用土耳其语转换规则来匹配该名称。但是，数据库服务器只有在知道对象是什么后才能告知要使用的转换规则，而它只有在知道要使用的转换规则之后才能知道对象是什么。无法为系统和用户对象正确解决此问题。在此示例中，由于数据库服务器使用的是土耳其语归类 1254TRK，因此第一条语句会失败，因为系统并不认为小写 **I** 等同于大写 **I**，而第二条语句则会成功。

土耳其语标准与 ANSI 标准的不兼容性要求土耳其语数据库中的系统对象引用必须以正确的大小写形式（即，对象创建时使用的大小写形式）指定对象名。应按如下方式编写上面的第一条语句：

```
SELECT * FROM SYSHISTORY
```

事实上，仅字母 **I** 必须采用正确的大小写形式。

或者，按如下方式编写该语句也是可以接受的，不过并不常见：

```
SELECT * FROM syshistory    //I-no-dot
```

请注意，诸如 INSERT 等关键字不区分大小写，即使在土耳其语数据库中也如此。SQL Anywhere 知道所有关键字均仅使用英语字母，所以它在匹配关键字时会使用 ANSI 大小写转换规则。SQL Anywhere 还会将此规律应用于其它一些标识符，如内置函数。不过，名称存储在目录中的对象必须使用正确的大小写形式或字母指定，如上所述。

不区分大小写的土耳其语数据库中的数据

在不区分大小写的土耳其语数据库中，也会对数据运用类似的规则。例如，如果数据值为

FIG

则对该数据的小写引用应为

fig

这样，同一 I-dot 字符就以两种形式使用。

替代土耳其语归类 1254TRKALT

对于一些应用程序开发人员，土耳其语字母 I 问题会导致严重的问题。正确的解决方案是确保所有对象引用都采用正确的大小写形式或使用字母 I 的正确形式，但是在某些情况下，做出违反土耳其语规则而符合 ANSI 规则的选择可能更为有利。

SQL Anywhere 提供了归类 1254TRKALT，除了使 I-dot 与 I-no-dot 成为等同字符外，该归类与 1254TRK 完全相同。

了解这种变化所带来的后果十分重要。在 1254TRKALT 数据库中，以下字符串是等同的：

fig

f1g

这对土耳其语用户而言不正确，但在一些情况下可能可以接受。

第二个问题发生在使用 ORDER BY 时。请看以下字符串：

ia

1a

1s

is

在 1254TRK 数据库中，对字符串进行 ORDER BY 会产生以下结果：

la
ls
ia
is

因为 I-no-dot 小于 I-dot。在 1254TRKALT 数据库中，顺序将是

ia
la
ls
is

因为 I-no-dot 等同于 I-dot。

管理用户 ID、特权和权限

目录

管理登录策略概述	416
数据库权限和特权概述	422
管理用户权限和特权概述	430
管理连接的用户	442
管理组	443
数据库对象名和前缀	450
使用视图和过程来获得更高的安全性	452
更改嵌套对象的所有权	454
如何确定用户权限	456
管理连接所使用的资源	457
目录中的用户和权限	458

管理登录策略概述

登录策略是数据库中的已命名对象，由一组在为某一用户创建数据库连接时应用的规则构成。所有新数据库都包含根登录策略。可以修改根登录策略的值，但不能删除该策略。登录策略仅控制适用于用户登录的规则，并与特权和权限分离。登录策略不能通过组成员资格继承。

登录策略可控制以下设置：

- 口令有效期
- 口令宽限期
- 下次登录时口令到期
- 已锁定
- 最大连接数
- 最大失败登录尝试数
- 最大已登录天数
- 最大非 DBA 连接数

在以下情况将根登录策略分配给用户帐户：

- 创建新用户帐户但没有指定登录策略
- 使用卸载实用程序 (dbunload) 重建上一版本 SQL Anywhere 创建的数据库
- 使用升级实用程序 (dbupgrad) 或 ALTER DATABASE UPGRADE 语句升级 SQL Anywhere 版本 10 的数据库

可以创建、修改和删除登录策略。同样，也可以创建、修改和删除用户，然后为用户分配登录策略。sa_get_user_status 系统过程允许获取有关某一用户当前状态的信息。请参见“sa_get_user_status 系统过程”一节《SQL Anywhere 服务器 - SQL 参考》。

继承登录策略设置

在数据库中存储了一个称为**根**的缺省登录策略，其中包含了可用于所有策略的缺省选项值。如果想使用缺省值以外的不同设置，可以修改根策略，或者创建一个策略然后对其进行更改，使其包含缺省值的替代值。如果没有修改策略使其包含替代值，策略将从根策略中继承缺省设置。

例如，假设 max_connections 的根策略值是 5。然后创建名为 myPolicy 的策略并对策略进行修改，以将 max_connections 的值设置为无限制。随后创建一个用户，并将 myPolicy 登录策略分配给该用户。当该用户登录时，除已设置为无限制的 max_connections 之外，其它登录策略选项设置都将从根登录策略中继承。

理解从根策略继承缺省值十分重要，因为如果随后变更根策略中某个选项设置的值，其登录策略依赖于该设置缺省值的所有用户都将受到影响。同样，即使更改了根值，但只要用户的策略包含该设置的替代值，该用户就不会受到任何影响。

修改根登录策略

◆ 修改根登录策略 (Sybase Central)

1. 以具有 DBA 特权的用户身份连接到数据库。
2. 在左窗格中，单击 [登录策略]。
3. 在右窗格中，右击 [根]，然后选择 [属性]。
4. 修改策略值并单击 [确定]。

◆ 修改根登录策略 (SQL)

1. 以具有 DBA 特权的用户身份连接到数据库。
2. 执行 ALTER LOGIN POLICY 语句。

示例

本示例在根登录策略中为已锁定值和 max_connections 值创建替代值。

```
ALTER LOGIN POLICY root
locked=on
max_connections=5;
```

另请参见

- “ALTER LOGIN POLICY 语句”一节 《SQL Anywhere 服务器 - SQL 参考》

创建新登录策略

如果还没有为用户分配所创建的登录策略，系统会为用户分配根策略。

◆ 创建登录策略 (Sybase Central)

1. 以具有 DBA 特权的用户身份连接到数据库。
2. 右击 [登录策略]，然后选择 [新建] » [登录策略]。
3. 请按照 [创建登录策略向导] 中的说明进行操作。

◆ 创建登录策略 (SQL)

1. 以具有 DBA 特权的用户身份连接到数据库。
2. 执行 CREATE LOGIN POLICY 语句。如果指定了已经存在的登录策略，语句失败。

示例

本示例创建具有选项值的登录策略 Test1。

```
CREATE LOGIN POLICY Test1;
```

另请参见

- “CREATE LOGIN POLICY 语句”一节 《SQL Anywhere 服务器 - SQL 参考》
- “为现有用户分配登录策略”一节第 418 页
- “修改登录策略”一节第 419 页

创建用户并为其分配登录策略

如果创建用户帐户后没有分配登录策略，系统会为其分配根登录策略。

◆ 创建新用户并为其分配登录策略 (Sybase Central)

1. 以具有 DBA 特权的用户身份连接到数据库。
2. 右击 [用户和组]，然后选择 [新建] » [用户]。
3. 请按照 [创建用户向导] 中的说明进行操作。

◆ 创建新用户并为其分配登录策略 (SQL)

1. 以具有 DBA 特权的用户身份连接到数据库。
2. 执行 CREATE USER 语句。

示例

本示例创建名为 SQLTester 的用户，其登录口令是 "welcome" 并为其分配了 Test1 登录策略。

```
CREATE USER SQLTester IDENTIFIED BY welcome  
LOGIN POLICY Test1;
```

另请参见

- “CREATE USER 语句”一节 《SQL Anywhere 服务器 - SQL 参考》
- “创建新用户”一节第 430 页

为现有用户分配登录策略

如果还没有为用户分配自定义登录策略，系统将为用户分配根登录策略。可使用此过程更改用户的登录策略分配。

◆ 为现有用户分配登录策略 (Sybase Central)

1. 以具有 DBA 特权的用户身份连接到数据库。
2. 在左窗格中，单击 [用户和组]。
3. 在右窗格中，右击某一用户然后选择 [属性]。
4. 在 [登录策略] 列表中选择一登录策略。

5. 单击 **[确定]**。

◆ 为现有用户分配登录策略 (SQL)

1. 以具有 DBA 特权的用户身份连接到数据库。
2. 执行 ALTER USER 语句。

示例

本示例将 Test2 登录策略分配给 SQLTester。

```
ALTER USER SQLTester  
LOGIN POLICY Test2;
```

另请参见

- “ALTER USER 语句”一节 《SQL Anywhere 服务器 - SQL 参考》
- “创建新用户”一节第 430 页

修改登录策略

◆ 修改登录策略 (Sybase Central)

1. 以具有 DBA 特权的用户身份连接到数据库。
2. 在左窗格中，单击 **[登录策略]**。
3. 在右窗格中，右击某一登录策略然后选择 **[属性]**。
4. 修改登录策略值。
5. 单击 **[确定]**。

◆ 修改登录策略 (SQL)

1. 以具有 DBA 权限的用户身份连接到数据库。
2. 执行 ALTER LOGIN POLICY 语句。

示例

本示例在 Test1 登录策略中为已锁定值和 max_connections 值创建替代值。

```
ALTER LOGIN POLICY Test1  
locked=on  
max_connections=5;
```

另请参见

- “ALTER LOGIN POLICY 语句”一节 《SQL Anywhere 服务器 - SQL 参考》
- “为现有用户分配登录策略”一节第 418 页

删除登录策略

根登录策略无法删除。在删除自定义登录策略前必须为用户分配另一登录策略。

注意

如果登录策略仍然分配给某个用户则不能将该策略删除。

◆ 删除登录策略 (Sybase Central)

1. 以具有 DBA 特权的用户身份连接到数据库。
2. 在左窗格中，单击 [登录策略]。
3. 在 [登录策略] 窗格中，右击某一登录策略然后选择 [删除]。
4. 单击 [是]。

◆ 删除登录策略 (SQL)

1. 以具有 DBA 特权的用户身份连接到数据库。
2. 执行 DROP LOGIN POLICY 语句。

示例

本示例删除登录策略 Test1。

```
DROP LOGIN POLICY Test1;
```

另请参见

- “DROP LOGIN POLICY 语句”一节 《SQL Anywhere 服务器 - SQL 参考》

在只读数据库中管理登录策略

当以只读模式启动数据库时，登录策略将以数据库的现有持久状态为基础。所分配的任何登录策略的影响仅限于当前会话。

如果在随后以只读模式启动的数据库上启用了登录管理，以下规则将适用：

- 服务器进行的登录管理以数据库启动前的状态为基础。
- 更改数据库状态的显式语句将被拒绝并导致错误。
- 服务器将继续维护各个用户的动态信息，如 failed_login_attempts 和 last_login_time。但是，服务器仅在瞬时内存中维护此信息，关闭数据库后这些信息将丢失。数据库将恢复为启动前的状态。
- 如果现有登录管理策略锁定了帐户，则用户无法登录。而且，通常用于在登录过程中更改口令的方法也无法使用。

- 在高可用性系统中，如果某一数据库由于充当镜像数据库而以只读模式工作，则在主数据库中执行的任何语句的影响都将在镜像数据库中有所反映。而且，在主服务器上收集的动态信息将被发送到镜像数据库，并在瞬时内存中与为镜像数据库收集的信息合并。

数据库权限和特权概述

数据库的每个用户都有一个在连接数据库时输入的名称（即用户 ID），并且用户至少属于一个组。用户和组也拥有赋予它们的特权和权限，从而使它们在数据库中维护信息安全性和保密性时能够执行各自的任務。

权限可赋予用户执行创建、修改、查询、使用或删除数据库对象（如表、视图、用户等等）任务的能力。**特权**可赋予用户在数据库级别执行任务的能力，例如备份数据库或执行诊断跟踪。SQL Anywhere 允许将权限和特权授予用户和组。

所有权限都是可继承的（从用户所属的组），但只有一部分特权是可继承的。

继承特权

下表列出了可分配给用户的特权，以及这些特权是否通过组成员资格继承：

特权	是否通过组成员资格继承	详细信息
BACKUP	否	请参见“BACKUP 特权”一节第 424 页。
DBA	否	请参见“DBA 特权”一节第 425 页。
PROFILE	是	请参见“PROFILE 特权”一节第 425 页。
READCLIENTFILE	是	请参见“READCLIENTFILE 特权”一节第 426 页。
READFILE	是	请参见“READFILE 特权”一节第 426 页。
REMOTE DBA		
RESOURCE	否	请参见“RESOURCE 特权”一节第 427 页。
VALIDATE	否	请参见“VALIDATE 特权”一节第 427 页。
WRITECLIENTFILE	是	请参见“WRITECLIENTFILE 特权”一节第 427 页。

继承权限

下表列出了可分配给用户的权限，以及这些权限是否通过组成员资格继承：

权限	是否通过组成员资格继承		详细信息
ALL	是	允许用户执行与数据库对象相关联的所有任务（相当于授予 ALTER、DELETE、INSERT、REFERENCES、SELECT 和 UPDATE 权限）	请参见“GRANT 语句”一节《SQL Anywhere 服务器 - SQL 参考》。
ALTER	是	允许用户修改数据库对象	请参见“通过组成员资格继承的权限”一节第 429 页。
CONNECT	否	允许用户连接到数据库	请参见“创建新用户”一节第 430 页。
CONSOLIDATE	否	在 SQL Remote 中标识统一数据库	请参见“CONSOLIDATE 权限”一节《SQL Remote》。
DELETE	是	允许用户删除数据库对象	请参见“GRANT 语句”一节《SQL Anywhere 服务器 - SQL 参考》。
INSERT	是	允许用户向数据库对象插入数据	请参见“GRANT 语句”一节《SQL Anywhere 服务器 - SQL 参考》。
INTEGRATED LOGIN	否	允许用户使用集成登录连接到数据库	请参见“使用集成登录”一节第 99 页。
KERBEROS LOGIN	否	允许用户使用 Kerberos 登录连接到数据库	请参见“Kerberos 验证”一节第 107 页。
PUBLISH	否	在 SQL Remote 中标识数据库的发布者	请参见“PUBLISH 权限”一节《SQL Remote》。
REFERENCES	是	允许用户在表上创建索引，并创建引用该表的外键	请参见“GRANT 语句”一节《SQL Anywhere 服务器 - SQL 参考》。

权限	是否通过组成员资格继承		详细信息
REMOTE	否	在 SQL Remote 和 MobiLink 中标识远程数据库	请参见“GRANT REMOTE DBA 语句 [MobiLink] [SQL Remote]”一节《SQL Anywhere 服务器 - SQL 参考》。
SELECT	是	允许用户查询数据库对象	请参见“GRANT 语句”一节《SQL Anywhere 服务器 - SQL 参考》。
UPDATE	是	允许用户更新数据库对象	请参见“GRANT 语句”一节《SQL Anywhere 服务器 - SQL 参考》。

另请参见：“GRANT 语句”一节《SQL Anywhere 服务器 - SQL 参考》。

负权限

SQL Anywhere 不支持负权限。这意味着无法撤除非显式授予的权限。

例如，假设用户 bob 是 sales 组的一个成员。如果某个用户向 sales 组授予对表 T 进行 DELETE 的权限，bob 就可以从 T 中删除行。如果您不想让 bob 从 T 中删除行，只是执行 REVOKE DELETE ON T 并不能撤消 bob 的 DELETE 权限，因为从未直接向 bob 授予过 DELETE ON T 权限。在此情况下，必须撤消 bob 在 sales 组中的成员资格。

请参见：

- “GRANT 语句”一节《SQL Anywhere 服务器 - SQL 参考》
- “REVOKE 语句”一节《SQL Anywhere 服务器 - SQL 参考》

特权概述

在 SQL Anywhere 中，可将**特权**看作是数据库级别的权限。因此，特权不必一定与数据库中的任何对象（用户除外）相关联。例如，拥有 BACKUP 特权的用户能够备份数据库。特权也可以包括数据库对象权限。例如，拥有 PROFILE 特权的用户可以执行应用程序分析和数据库跟踪任务，这需要其他用户（具有 DBA 特权的用户除外）无法使用的系统表和系统过程。

BACKUP 特权

BACKUP 特权允许用户使用 BACKUP 语句或 dbbackup 实用程序以档案备份或映像备份的方式来备份数据库和事务日志。BACKUP 特权不能通过组成员资格继承，只能由具有 DBA 特权的用户授予。请参见“BACKUP 语句”一节《SQL Anywhere 服务器 - SQL 参考》和“备份实用程序 (dbbackup)”一节第 698 页。

DBA 特权

在您创建数据库时，一个可用用户 ID 也会随之创建。缺省情况下，第一个用户 ID 为 **DBA**，初始口令为 **sql**（口令区分大小写）。您可以使用 **CREATE DATABASE** 语句的 **DBA USER** 和 **DBA PASSWORD** 子句或通过指定 **dbinit -dba** 选项来更改 DBA 用户的名称和口令。请参见“[CREATE DATABASE 语句](#)”一节《[SQL Anywhere 服务器 - SQL 参考](#)》和“[初始化实用程序 \(dbinit\)](#)”一节第 729 页。

DBA 用户 ID 在数据库中自动拥有 DBA 特权。有了此级别的权限，DBA 用户可以在数据库中执行任何任务。他们可以创建表、更改表结构、创建新用户 ID、撤消用户的权限、备份数据库，等等。

DBA 特权不能通过组成员资格继承。

仅在您连接到数据库时，才对数据库拥有 DBA 特权。

拥有 DBA 特权的用户

拥有 DBA 特权的用户将成为数据库管理员。所指的数据库管理员或 DBA 包括拥有 DBA 特权的任何用户。

虽然 DBA 特权可以被授予或转让给其它用户 ID，但本章还是假定 DBA 用户 ID 是数据库管理员，而 DBA 这一缩写既表示 DBA 用户 ID，也表示任何拥有 DBA 特权的用户 ID。

添加新用户

DBA 拥有向数据库添加新用户的特权。当 DBA 添加用户时，同时会授予这些用户在数据库中执行任务的权限。有些用户可能只需要使用 SQL 查询来查看数据库信息，而其他一些用户则可能需要向数据库添加信息，另外还有一些用户可能会需要修改数据库本身的结构。虽然 DBA 的某些职责可以移交给其它用户 ID，但由于 DBA 具有 DBA 特权，因而仍由它来负责数据库的全面管理。

DBA 具有创建数据库对象和为其它用户 ID 指派这些对象的所有权的特权。

小心

为防止对数据进行未经授权的访问，您应在部署数据库之前更改 DBA 用户的口令（或更改 DBA 用户和口令）。

PROFILE 特权

PROFILE 特权允许用户执行以下分析、跟踪和诊断操作：

- 应用程序分析
- 诊断跟踪
- 过程分析
- 请求记录（创建并分析请求日志）
- 运行索引顾问

PROFILE 特权不能通过组成员资格继承。

拥有 PROFILE 特权的用户将被自动添加到诊断组中。

执行过程分析和请求记录时，用户需要使用 `sa_server_option` 系统过程来进行配置。对此过程的访问权限将被授予拥有 `PROFILE` 特权的用户，但仅限于那些与过程分析和请求记录相关联的选项。

当执行应用程序分析和诊断跟踪时，拥有 `PROFILE` 特权但不具有 `DBA` 特权的用户无法创建单独的数据库来存储分析和跟踪数据，除非他们拥有卸载数据库的特殊权限。但是，他们可以将数据存储在当前数据库或已成功连接的另一数据库中。

要授予此特权，数据库必须由 SQL Anywhere 11 的数据库服务器创建，或者已使用升级实用程序 (`dbupgrad`) 或 `ALTER DATABASE UPGRADE` 语句将数据库升级到版本 11。请参见“[升级实用程序 \(dbupgrad\)](#)”一节第 808 页和“[ALTER DATABASE 语句](#)”一节《[SQL Anywhere 服务器 - SQL 参考](#)》。

另请参见

- “应用程序分析”一节《[SQL Anywhere 服务器 - SQL 的用法](#)》
- “使用诊断跟踪进行高级应用程序分析”一节《[SQL Anywhere 服务器 - SQL 的用法](#)》
- “索引顾问”一节《[SQL Anywhere 服务器 - SQL 的用法](#)》
- “使用系统过程进行过程分析”一节《[SQL Anywhere 服务器 - SQL 的用法](#)》
- “请求记录”一节《[SQL Anywhere 服务器 - SQL 的用法](#)》
- “DBA 特权”一节第 425 页
- “`sa_server_option` 系统过程”一节《[SQL Anywhere 服务器 - SQL 参考](#)》

READCLIENTFILE 特权

`READCLIENTFILE` 特权允许用户读取客户端计算机上的文件，例如从客户端计算机上的文件装载数据时。

`READCLIENTFILE` 特权可以通过组成员资格继承。

另请参见

- “访问客户端计算机上的数据”一节《[SQL Anywhere 服务器 - SQL 的用法](#)》
- “`WRITECLIENTFILE` 特权”一节第 427 页
- “`GRANT` 语句”一节《[SQL Anywhere 服务器 - SQL 参考](#)》
- “`LOAD TABLE` 语句”一节《[SQL Anywhere 服务器 - SQL 参考](#)》
- “`READ_CLIENT_FILE` 函数 [String]”一节《[SQL Anywhere 服务器 - SQL 参考](#)》

READFILE 特权

`READFILE` 特权允许用户在 `SELECT` 语句中使用 `OPENSTRING` 子句来读取文件。如果没有 `READFILE` 特权，用户仍然可以使用 `OPENSTRING` 子句查询字符串或 `BLOB` 值，但无法查询文件。

`READFILE` 特权可以通过组成员资格继承。

有关在 `SELECT` 语句中使用 `OPENSTRING` 子句的详细信息，请参见“[FROM 子句](#)”一节《[SQL Anywhere 服务器 - SQL 参考](#)》。

REMOTE DBA 特权

REMOTE DBA 特权会向 SQL Remote 或 MobiLink 同步用户授予一组有限的 DBA 权限。远程 DBA 特权避免了必须授予完全 DBA 特权，从而避免了与分发 DBA 用户 ID 和口令相关的安全问题。

有关使用 REMOTE DBA 特权的详细信息，请参见“[GRANT REMOTE DBA 语句 \[MobiLink \[SQL Remote\]\]](#)”一节《[SQL Anywhere 服务器 - SQL 参考](#)》。

RESOURCE 特权

RESOURCE 特权允许用户创建数据库对象（例如表、视图、存储过程和触发器）。RESOURCE 特权不能通过组成员资格继承，只能由拥有 DBA 特权的用户授予。

要创建触发器，用户需要 RESOURCE 特权和应用触发器的表的 ALTER 权限。

创建具有不同所有者的数据库对象需要 DBA 特权。

VALIDATE 特权

VALIDATE 特权允许用户使用 VALIDATE 语句或 dbvalid 实用程序校验数据库、表、索引和校验和。VALIDATE 特权不能通过组成员资格继承，只能由拥有 DBA 特权的用户授予。

另请参见

- “[VALIDATE 语句](#)”一节《[SQL Anywhere 服务器 - SQL 参考](#)》
- “[校验实用程序 \(dbvalid\)](#)”一节第 810 页

WRITECLIENTFILE 特权

WRITECLIENTFILE 特权允许用户对客户端计算机上的文件执行写入操作，例如使用 UNLOAD TABLE 语句向客户端计算机写入数据时。

WRITECLIENTFILE 特权可以通过组成员资格继承。

另请参见

- “[访问客户端计算机上的数据](#)”一节《[SQL Anywhere 服务器 - SQL 的用法](#)》
- “[READCLIENTFILE 特权](#)”一节第 426 页
- “[GRANT 语句](#)”一节《[SQL Anywhere 服务器 - SQL 参考](#)》
- “[READ_CLIENT_FILE 函数 \[String\]](#)”一节《[SQL Anywhere 服务器 - SQL 参考](#)》
- “[WRITE_CLIENT_FILE 函数 \[String\]](#)”一节《[SQL Anywhere 服务器 - SQL 参考](#)》
- “[UNLOAD 语句](#)”一节《[SQL Anywhere 服务器 - SQL 参考](#)》

权限概述

在 SQL Anywhere 中，权限允许用户访问、创建、修改以及删除数据库对象（表、视图、过程，等等）。例如，要从表中选择数据，则用户必须拥有该表，或者必须具有对该表的 SELECT 权限。

用户权限可分为以下几个主要类别：

- **为用户或组显式设置的权限** 为用户或组显式设置的权限，用于控制用户创建、修改、执行或删除数据库对象的能力。
- **由于拥有某一对象而获得的权限** 由于创建数据库对象而获得的相应权限。例如，如果用户创建了表，则对表的所有权允许用户修改或删除该对象。
- **通过组成员资格继承的权限** 从用户或组所属的组继承的权限。
- **对禁用对象的权限** 可以授予对禁用对象的权限。对禁用对象的权限存储在数据库中，在对象被启用时生效。

为用户或组显式设置的权限

可以赋予用户执行系统过程和函数的权限，方法是授予用户此类对象的 EXECUTE 权限。

对于表、视图和 dbspace，可向用户 ID 授予几种不同的权限：

权限	说明
ALTER	变更表的结构或在表上创建触发器的权限。
CREATE ON	在指定的 dbspace 上创建数据库对象的权限。
DELETE	从表或视图中删除行的权限。
INSERT	在表或视图中插入行的权限。
REFERENCES	在表上创建索引以及创建引用表的外键的权限。
SELECT	在表或视图中查看信息的权限。
UPDATE	更新表或视图中行的权限。还能针对表或视图中的一组列授予此权限。
ALL	所有上述权限。

有关可对数据库对象设置的权限的详细信息，请参见“GRANT 语句”一节《SQL Anywhere 服务器 - SQL 参考》。

由于拥有某一对象而获得的权限

在数据库中创建新对象的用户被称为该对象的**所有者**，且该用户自动拥有对该对象执行任何操作的权限。例如，表的所有者可以修改表的结构，也可向其他数据库用户授予更新表中信息的权限。

拥有 DBA 特权的用户有修改数据库中任何组成部分的权限，因此可以删除其他用户所创建的表。他们拥有各个数据库对象所有者所拥有的有关该对象的全部权限。此外，拥有 DBA 特权的用户也可以为其他用户创建数据库对象。在这种情况下，对象的所有者不是执行 CREATE 语句的用户 ID。尽管存在这种可能性，但在提到数据库对象的所有者和创建者时，都会将其视为同一用户，并且二者可以互换。

另请参见

- “没有口令的组”一节第 447 页

通过组成员资格继承的权限

为数据库的各个用户分别设置权限可能非常耗时且容易出错。对于多数数据库，以组为单位（而不是针对单个用户 ID）对权限进行管理会更加有效。

各个用户 ID 都可以是多个组的成员，而且它们继承各个组的各种权限。

例如，您可以在公司数据库中为不同的部门（销售部、市场部等）分别创建不同的组并为这些组指派权限。每一名销售人员都会成为销售组的成员，并自动获得访问数据库相应区域的权限。

管理用户权限和特权概述

本节将介绍如何创建新用户并为其授予权限和特权。对于多数数据库，应使用**组**来执行批量权限管理，而不是为各用户一次指派一个权限。但是，由于组只是一个具有特殊属性的用户 ID，所以在继续阅读有关管理组的论述之前，您应该先阅读并理解本节的内容。

设置单个用户 ID

即使不存在有关多用户数据库的安全问题，也还是有必要为各个用户设置单独的用户 ID。除了向各用户授予权限外，您还可以向用户组授予权限。如果建立具有适当权限的组，管理开销会很低。您可能需要使用单个的用户 ID，这是因为：

- 翻译日志文件实用程序 (dblog) 可以从事务日志中有选择性地抽取各用户所进行的更改。这对于在数据不正确时进行故障排除或弄清所发生的事件非常有用。
- Sybase Central 显示了更加有用的信息，以便您能够分辨出哪些连接属于哪些用户。
- 行锁定消息 (blocking 选项被设置为 Off 时) 可提供更丰富的信息。

创建新用户

在 Sybase Central 和 Interactive SQL 中都可以创建新用户。在 Sybase Central 中，您可以在 [用户和组] 文件夹中管理用户或组。在 Interactive SQL 中，您可以使用 CREATE USER 语句添加新用户。要想在这两种工具中创建新用户，您都需要有 DBA 特权。

所有新用户都将自动添加到 PUBLIC 组中。在创建了新用户后，您可以：

- 将用户添加到其它组。请参见“[将组成员资格授予现有的用户或组](#)”一节第 444 页。
- 设置用户对表、视图和过程的操作权限。请参见“[管理用户权限和特权概述](#)”一节第 430 页。
- 将用户设置为发布者或数据库的远程用户。请参见“[用户权限](#)”一节《[SQL Remote](#)》。
- 向用户指派登录策略。缺省情况下,将向用户指派根登录策略。然而，您可以创建并指派自定义登录策略。请参见“[管理登录策略概述](#)”一节第 416 页。

新用户的初始权限

缺省情况下，分配给新用户的权限包括：

- 连接到数据库的权限（假定用户口令已指定）
- 查看存储在系统视图中数据的权限
- 执行大多数系统存储过程的权限

如果新用户要访问数据库中的表，需要给他们指派权限。

通过向特殊的 PUBLIC 用户组指派权限，拥有 DBA 特权的用户可以设置自动授予新用户的权限。请参见“[特殊组](#)”一节第 448 页。

对用户 ID 和口令的限制

用户 ID 不能:

- 以空格、单引号或双引号开头
- 以空格结尾
- 含有分号

口令区分大小写并且不能:

- 以空格、单引号或双引号开头
- 以空格结尾
- 含有分号
- 长度超过 255 个字节

请参见“设置口令”一节第 431 页。

◆ 创建新用户 (Sybase Central):

1. 以具有 DBA 特权的用户身份连接到数据库。
2. 右击 [用户和组], 然后选择 [新建] » [用户]。
3. 请按照 [创建用户向导] 中的说明进行操作。

◆ 创建新用户 (SQL):

1. 以具有 DBA 特权的用户身份连接到数据库。
2. 执行 CREATE USER 语句。

示例

本示例向数据库添加一名新用户, 用户 ID 为 M_Haneef, 口令为 Welcome。

```
CREATE USER M_Haneef  
IDENTIFIED BY Welcome;
```

另请参见

- “CREATE USER 语句”一节 《SQL Anywhere 服务器 - SQL 参考》

设置口令

用户必须拥有口令才能连接到数据库。口令区分大小写并且不能:

- 以空格、单引号或双引号开头
- 以空格结尾
- 含有分号
- 长度超过 255 个字节

创建或更改口令时，口令先被转换为 UTF-8，然后再散列并存储在数据库中。如果数据库被卸载和重装到带有不同字符集的数据库，则现有口令将继续有效。如果服务器不能将客户端的字符集转换为 UTF-8，建议采用由 7 位 ASCII 字符组成的口令，因为其它字符可能会无法正常工作。

更改口令

◆ 更改用户口令 (Sybase Central)

1. 以具有 DBA 特权的用户身份连接到数据库。
2. 单击 [用户和组]。
3. 在 [用户和组] 列表中，右击某一用户然后选择 [属性]。
4. 选择 [此用户具有口令]。
5. 填写 [口令] 和 [确认口令] 字段。
6. 单击 [应用]。
7. 单击 [确定]。

◆ 更改用户口令 (SQL)

1. 以具有 DBA 权限的用户身份连接到数据库。
2. 执行 ALTER USER 语句。

示例

```
ALTER USER M Haneef  
IDENTIFIED BY welcome;
```

更改 DBA 口令

所有数据库的 DBA 用户的缺省口令都是 sql。您应更改此口令以防止对数据库的非授权访问。

◆ 更改 DBA 口令 (Sybase Central)

1. 以具有 DBA 权限的用户身份连接到数据库。
2. 单击 [用户和组]。
3. 在 [用户和组] 列表中，右击 [DBA] 然后选择 [属性]。
4. 选择 [此用户具有口令]。
5. 填写 [口令] 和 [确认口令] 字段。
6. 单击 [应用]。
7. 单击 [确定]。

◆ 更改 DBA 口令 (SQL)

1. 以具有 DBA 权限的用户身份连接到数据库。
2. 执行 ALTER USER 语句。

示例

以下命令将 DBA 用户的口令更改为 welcome_DBA:

```
ALTER USER DBA  
IDENTIFIED BY welcome_DBA;
```

另请参见

- “ALTER USER 语句”一节 《SQL Anywhere 服务器 - SQL 参考》

设置用户和组选项

在 Sybase Central 中，用户和组的可配置选项位于 [用户选项] 和 [组选项] 窗口（与设置数据库选项的窗口相同）中。在 Interactive SQL 中，您可在 SET OPTION 语句中指定选项。

◆ 为用户或组设置选项 (Sybase Central):

1. 以具有 DBA 特权的用户身份连接到数据库。
2. 单击 [用户和组]。
3. 右击用户或组，然后选择 [选项]。
4. 在 [选项] 列表中单击一个选项。
5. 单击 [立即设置永久值]。
6. 单击 [关闭]。

◆ 为用户或组设置选项 (SQL):

1. 以具有 DBA 权限的用户身份连接到数据库。
2. 执行 SET OPTION 语句。

另请参见

- “设置数据库对象的属性”一节 《SQL Anywhere 服务器 - SQL 的用法》
- “数据库选项”一节第 466 页

授予特权

授予特权的方式与授予权限的方式大致相同。

◆ 向用户 ID 授予特权

1. 以具有 DBA 特权的用户身份连接到数据库。
2. 执行 GRANT 语句，指定要授予的特权和接收者的用户 ID。

例如，要授予 DBA 特权，正确的 SQL 语句为：

```
GRANT DBA TO user-id;
```

有关 SQL Anywhere 中支持的特权的详细信息，请参见“[特权概述](#)”一节第 424 页。

授予针对表的权限

您可以对单个表指派一组权限，并向用户授予这些权限组合，以规定他们对表的访问权限。

您可以使用 Sybase Central 或 Interactive SQL 来设置权限。在 Interactive SQL 中，可以使用 GRANT 语句来授予以下针对表的权限：

- ALTER 权限允许用户变更表的结构或在表中创建触发器。REFERENCES 权限允许用户在表中创建索引和创建外键。这些权限可授予修改数据库模式的特权，因此不会将这些权限指派给大多数用户。这些权限不适用于视图。
- DELETE、INSERT 和 UPDATE 权限授予修改表中数据的特权。
- SELECT 权限可授予在表中查看数据的特权，但不授予更改数据的权限。
- ALL 权限可授予所有上述权限。
- REFERENCES、SELECT 和 UPDATE 权限可以被限制到表或视图中的一组列。

◆ 授予针对表或列的权限 (Sybase Central):

1. 以具有 DBA 特权的用户身份连接到数据库。
2. 单击 [表]。
3. 右击某个表，然后选择 [属性]。
4. 单击 [权限] 选项卡，然后配置该表的权限：
 - 单击 [授权]。
 - 双击某一用户或组。
 - 在权限表中，单击该用户或组旁边的字段以设置特定权限。
 - 选择某个用户然后单击 [更改] 来设置对列的特定权限。
 - 单击 [确定]。
 - 要撤消所有权限，请选择某个用户或组并单击 [撤消]。
5. 单击 [应用]。

提示

也可以在 [用户属性] 或 [组属性] 窗口中指派权限。要向多个用户或组指派权限，请使用 [表属性] 窗口。要向多个表指派权限，请使用 [用户属性] 窗口。

◆ 授予针对表或列的权限 (SQL):

1. 以具有 DBA 特权的用户身份连接到数据库。
2. 执行 GRANT 语句来指派权限。

请参见“GRANT 语句”一节《SQL Anywhere 服务器 - SQL 参考》。

示例 1

所有表权限都以非常相似的方式授予。您可为 M_Haneef 授予从名为 sample_table 的表中删除行的权限，方法如下：

1. 以拥有 DBA 特权的用户身份或 sample_table 的所有者身份连接到数据库。
2. 执行以下 SQL 语句：

```
GRANT DELETE
ON sample_table
TO M_Haneef;
```

示例 2

您可以为 M_Haneef 授予仅在名为 sample_table 的表中更新 column_1 和 column_2 列的权限，方法如下：

1. 以拥有 DBA 特权的用户身份或 sample_table 的所有者身份连接到数据库。
2. 执行以下 SQL 语句：

```
GRANT UPDATE ( column_1, column_2 )
ON sample_table
TO M_Haneef;
```

对表进行权限设置有其局限性，即表权限通常应用于表中的所有数据，虽然可以针对列的一个子集授予 REFERENCES、SELECT 和 UPDATE 权限。为了调整用户权限，您可以创建对表执行操作的过程，然后把执行该过程的权限授予用户。

另请参见

- “GRANT 语句”一节《SQL Anywhere 服务器 - SQL 参考》

授予针对视图的权限

设置针对视图的权限与设置针对表的权限类似。

有关所涉及的 SQL 语句的详细信息，请参见“授予针对表的权限”一节第 434 页。

如果具备以下条件中的一个和多个，用户可以通过视图执行操作：

- 拥有 DBA 特权的用户已授予该用户对视图执行适当操作的权限。
 - 该用户拥有对所有基表执行相应操作的权限。
 - 一个非 DBA 用户为该用户授予了对视图进行相应操作的权限。该非 DBA 用户必须是视图的所有者，或者必须具有对视图的适当权限的授权选项。视图的所有者必须是以下之一：
 - 具有 DBA 特权的用户。
 - 没有 DBA 特权但拥有该视图所引用的所有基表的用户。
 - 没有 DBA 特权并且也不是该视图所引用的部分或全部基表的所有者的用户。但是，该视图所有者具备针对非其拥有的基表的 SELECT 权限及授权选项 (WITH GRANT OPTION)，以及操作非其拥有的基表所需的任何其它权限和授权选项 (WITH GRANT OPTION)。
- 权限可能已被授予 PUBLIC，而不是对基表具有权限 WITH GRANT OPTION 的所有者。其中包括针对系统表的 SELECT 权限。

可以对整个视图或视图中的各列授予 UPDATE 权限。

注意

可以授予对禁用视图的权限。对禁用视图的权限存储在数据库中，在对象被启用时生效。

◆ **授予针对视图的权限 (Sybase Central):**

1. 以具有 DBA 特权的用户身份连接到数据库。
2. 单击 [视图]。
3. 右击某个视图，然后选择 [属性]。
4. 单击 [权限] 选项卡。
5. 配置针对视图的权限：
 - 单击 [授权]。
 - 双击某一用户或组。
 - 在权限表中，单击该用户或组旁边的字段以设置特定权限。
 - 要撤消所有权限，请选择某个用户或组并单击 [撤消]。
6. 单击 [应用]。

提示

也可以在 [用户属性] 或 [组属性] 窗口中指派权限。使用 [视图属性] 窗口可向多个用户或组指派权限。使用 [用户属性] 或 [组属性] 窗口可向多个视图指派权限。

另请参见

- “GRANT 语句”一节 《SQL Anywhere 服务器 - SQL 参考》

授予用户授予权限的权限

您可以使用 WITH GRANT OPTION 来指派所描述的各个表权限和视图权限。该选项授予将权限传递给其他用户的权利。

在 Sybase Central 中，您可以指定授权选项。方法是：显示用户、组或表的属性窗口，单击 [权限] 选项卡，然后在所提供的字段中双击，这样，就会出现带有两个 '+' 号的选中标记。

注意

您只能为用户指定 WITH GRANT OPTION。如果将 WITH GRANT OPTION 授予组，则组的成员不会继承 WITH GRANT OPTION。

示例

您可以给 M_Haneef 授予从名为 sample_table 的表中删除行的权限，以及将此权限传递给其他用户的权限，方法如下：

1. 以拥有 DBA 特权的用户身份或 sample_table 的所有者身份连接到数据库
2. 执行以下 SQL 语句：

```
GRANT DELETE ON sample_table  
TO M_Haneef  
WITH GRANT OPTION;
```

请参见“GRANT 语句”一节《SQL Anywhere 服务器 - SQL 参考》。

另请参见

- “授予针对表的权限”一节第 434 页
- “组的权限和特权”一节第 446 页

授予针对过程的权限

具有 DBA 特权的用户或过程的所有者可以授予执行存储过程的权限。可针对过程授予的唯一权限是 EXECUTE 权限。

授予执行过程权限的方法与授予针对表和视图的权限的方法相似。但是，GRANT 语句的 WITH GRANT OPTION 子句不适用于授予针对过程的权限。

您可以使用 Sybase Central 或 Interactive SQL 来设置权限。

◆ 授予针对过程的权限 (Sybase Central):

1. 以具有 DBA 特权的用户身份连接到数据库。
2. 单击 [过程和函数]。
3. 右击某个过程，然后选择 [属性]。
4. 单击 [权限] 选项卡。

5. 配置针对过程的权限:

- 单击 [授权]。
- 双击某一用户或组。
- 要允许或撤消执行过程的权限, 请选择某个用户或组并单击 [执行] 列。可以执行该过程的用户或组将以复选标记表示。
- 要撤消所有权限, 请选择某个用户或组并单击 [撤消]。

6. 单击 [应用]。

提示

也可以在 [用户属性] 或 [组属性] 窗口中指派权限。使用 [过程属性] 窗口可向多个用户或组指派权限。使用 [用户属性] 或 [组属性] 窗口可向多个过程指派权限。

◆ **授予针对过程的权限 (SQL):**

1. 作为具有 DBA 特权的用户或作为过程的所有者连接到数据库。
2. 执行 GRANT EXECUTE ON 语句。

示例

您可以授予 M_Haneef 执行名为 my_procedure 的过程的权限, 方法如下:

1. 作为具有 DBA 特权的用户或作为 my_procedure 过程的所有者连接到数据库。
2. 执行以下 SQL 语句:

```
GRANT EXECUTE  
ON my_procedure  
TO M_Haneef;
```

过程的执行权限

使用过程所有者的权限执行过程。仅当过程所有者拥有针对表的 UPDATE 权限时, 更新表中信息的任何过程才能成功执行。

只要过程所有者拥有适当权限, 当拥有执行该过程的权限的用户调用该过程时, 无论该用户是否拥有针对基表的权限, 该过程均可成功执行。您可以使用过程来允许用户对表执行明确定义的活动, 而不必拥有针对表的任何常规权限。

另请参见

- “GRANT 语句”一节 《SQL Anywhere 服务器 - SQL 参考》
- “授予针对表的权限”一节第 434 页

触发器的执行权限

服务器通过执行触发器来响应用户操作。执行触发器不需要权限。当触发器执行时, 它会使用与其相关联的表的创建者的权限。

请参见“触发器执行权限”一节《SQL Anywhere 服务器 - SQL 的用法》。

授予和撤消远程权限

您可以在 Sybase Central 中管理用户和组的远程权限。远程权限允许普通用户和组成为 SQL Remote 复制设置中的远程用户，以便与发布数据库交换复制消息。

授予远程权限

您至少要先在数据库中定义一个消息类型，然后您才能授予用户远程权限。

要为组授予远程权限，必须为组中的各个用户显式授予远程权限。组成员不继承远程权限。

撤消远程权限

撤消远程权限会使远程用户恢复为普通用户。撤消这些权限也将自动从所有发布中取消该用户的预订。

◆ 给用户授予远程权限 (Sybase Central):

1. 以具有 DBA 权限的用户身份连接到数据库。
2. 单击 [用户和组]。
3. 右击某一用户，然后选择 [更改为远程用户]。
4. 填写字段，然后单击 [确定]。

向用户授予远程权限后，可以为该用户预订发布。

◆ 撤消远程用户的远程权限:

1. 以具有 DBA 特权的用户身份连接到数据库。
2. 单击 [用户和组] 或 [SQL Remote 用户]。
3. 右击某一用户，然后选择 [撤消远程]。
4. 单击 [是]。

请参见“SQL Remote 简介”《SQL Remote》。

撤消用户权限和特权

用户的权限是已授予和已撤消权限的组合。通过撤消和授予权限，您可以管理用户对于数据库所拥有的权限的模式。

必须由具有 DBA 特权的用户或过程的所有者发出此命令。

如果正撤消其他用户的连接权限或表权限，则此时该用户不能与数据库相连。无法撤消 dbo 的连接权限。

REVOKE 语句可撤消已被显式授予给用户的权限（即不是从用户所属的组中继承的权限）。REVOKE 语句的语法与 GRANT 语句的语法相同。例如，要撤消用户 M_Haneef 执行 my_procedure 的权限，则命令为：

```
REVOKE EXECUTE
ON my_procedure
FROM M_Haneef;
```

要撤消用户从 sample_table 删除行的权限，命令为：

```
REVOKE DELETE
ON sample_table
FROM M_Haneef;
```

将用户添加到组中时，该用户将继承指派给该组的所有权限以及可继承的特权。SQL Anywhere 不允许撤消用户作为一个组的成员而继承的权限和特权子集。只可撤消由 GRANT 语句显式授予的权限。如果需要移除用户继承的权限或特权，可以考虑创建一个具有所需权限和特权的新组并使用户成为其中的一员，或者先将用户从组中移除，然后再显式授予该用户所需的权限。

另请参见

- “REVOKE 语句”一节 《SQL Anywhere 服务器 - SQL 参考》
- “GRANT 语句”一节 《SQL Anywhere 服务器 - SQL 参考》

从数据库删除用户

使用 Sybase Central 和 Interactive SQL 都可以从数据库中删除用户。在此过程中，正在删除的用户不能连接到该数据库。

删除用户之后，也将删除该用户拥有的所有数据库对象（例如表）。

只有具有 DBA 特权的用户才能删除用户。

◆ 从数据库中删除用户 (Sybase Central):

1. 以具有 DBA 特权的用户身份连接到数据库。
2. 单击 [用户和组]。
3. 右击某一用户，然后选择 [删除]。
4. 单击 [是]。

◆ 从数据库中删除用户 (SQL):

1. 以具有 DBA 特权的用户身份连接到数据库。
2. 执行 DROP USER 语句。

示例

从数据库中移除用户 M_Haneef。

```
DROP USER M_Haneef;
```

另请参见

- “DROP USER 语句”一节 《SQL Anywhere 服务器 - SQL 参考》
- “REVOKE 语句”一节 《SQL Anywhere 服务器 - SQL 参考》
- “撤消用户权限和特权”一节第 439 页
- “从数据库删除组”一节第 448 页

管理连接的用户

如果使用 Sybase Central，您可以跟踪连接到数据库的所有用户。您可以查看这些已连接的用户属性，并且可以在需要时断开他们的连接。

◆ 显示连接到数据库的所有用户的列表：

- 在左窗格中选择数据库，然后单击右窗格中的 [连接的用户] 选项卡。

此选项卡将显示当前连接到数据库的所有用户，而不管他们用来进行连接的应用程序是什么（Sybase Central、Interactive SQL 或自定义客户端应用程序）。

◆ 检查连接到数据库的用户的属性：

1. 在左窗格中选择数据库，然后单击右窗格中的 [连接的用户] 选项卡。
2. 右击该用户然后选择 [属性]。
3. 审查该用户的属性，然后单击 [确定]。

◆ 断开用户与数据库的连接：

1. 在左窗格中选择数据库，然后单击右窗格中的 [连接的用户] 选项卡。
2. 右击该用户，然后选择 [断开连接]。

管理组

可将组视为拥有特殊权限（如可以具有成员）的用户 ID。授予和撤消组权限和特权的方式与对用户的处理方式完全相同。

您可以构建若干个组的层次，其中每个组都是另一个组的成员。无论成员是用户还是组，其都会继承父组的特权和权限。每个用户 ID 可以属于多个组；用户与组的关系是多对多的关系。

正如同用户一样，您可以在表、视图或过程中授予或撤消组权限。当您如此操作时，组的所有成员均会继承更改。

您可以创建没有口令的组，这样可防止用户使用组用户 ID 连接到数据库。请参见“[没有口令的组](#)”一节第 447 页。

要管理组的特权和权限，请执行与管理用户的权限和特权相同的过程。请参见“[管理用户权限和特权概述](#)”一节第 430 页。

要管理组的远程权限，请参见“[授予和撤消远程权限](#)”一节第 439 页。

组继承的特殊注意事项

除了授予权限（GRANT ...WITH GRANT OPTION 语句）外，用户和组均继承其所在组的全部权限。

组成员可以只继承其所属组的以下特权集。

- READCLIENTFILE
- READFILE
- WRITECLIENTFILE

简要示例

在以下示例中创建了两个组：group1 和 group2。还创建了一个用户 bobsmith 并赋予其在两个组中的成员资格。此外，又创建了一个表 table1，并赋予 group2 对该新表的 SELECT 和 INSERT 权限。

```
GRANT CONNECT, GROUP TO group1;
GRANT CONNECT, GROUP TO group2;
GRANT CONNECT TO bobsmith IDENTIFIED BY sql;
GRANT MEMBERSHIP IN GROUP group1 TO bobsmith;
GRANT MEMBERSHIP IN GROUP group2 TO bobsmith;

CREATE TABLE DBA.table1( column1 INT, modified_by VARCHAR(128) DEFAULT
USER );
GRANT SELECT, INSERT ON DBA.table1 TO group2;
```

因为 bobsmith 是 group2 的成员，所以他继承了对 table1 的选择和插入权限，可插入如下所示的值：

```
CONNECT USER bobsmith IDENTIFIED BY sql;
INSERT INTO DBA.table1(column1) VALUES(1);
```

创建组

◆ 创建新组 (Sybase Central):

1. 以具有 DBA 特权的用户身份连接到数据库。
2. 右击 [用户和组] 然后选择 [新建] » [组]。
3. 请按照 [创建组向导] 中的说明进行操作。

◆ 创建新组 (SQL):

1. 以具有 DBA 特权的用户身份连接到数据库。
2. 执行 GRANT GROUP TO 语句。如果在此语句中指定的用户 ID 尚未创建，则该语句失败。

示例

创建用户 ID personnel。

```
CREATE USER personnel  
IDENTIFIED BY group_password;
```

使用户 ID personnel 成为组。

```
GRANT GROUP TO personnel;
```

另请参见

- “GRANT 语句”一节 《SQL Anywhere 服务器 - SQL 参考》
- “CREATE USER 语句”一节 《SQL Anywhere 服务器 - SQL 参考》
- “创建新用户”一节第 430 页

将组成员资格授予现有的用户或组

在 Sybase Central 和 Interactive SQL 中，您都可以将现有用户添加到组，或将组添加到其它组。在 Sybase Central 中，您可以在用户或组的右窗格中控制组成员资格。在 Interactive SQL 中，可以使用 GRANT 语句使用户成为组成员。

当您在组中给用户指派成员资格时，它们会继承与该组相关的所有针对表、视图和过程的权限。同时他们也将继承所有可继承的特权。

只有具有 DBA 特权的用户才能授予组成员资格。

◆ 将用户或组添加到另一个组 (Sybase Central):

1. 以具有 DBA 特权的用户身份连接到数据库。
2. 单击 [用户和组]。
3. 双击某一用户或组。

4. 对于用户：
 - 从 [文件] 菜单选择 [新建] » [成员资格]。对于组：
 - 单击 [成员资格] 选项卡。
 - 从 [文件] 菜单选择 [新建] » [成员资格]。
5. 在 [名称] 列表中双击某个组。

新组将出现在右窗格中的 [成员资格] 选项卡上。

◆ 将用户或组添加到另一个组 (SQL)

1. 以具有 DBA 特权的用户身份连接到数据库。
2. 执行 GRANT MEMBERSHIP IN GROUP 语句，指定组和涉及的用户。

示例

在 personnel 组中授予用户 M_Haneef 成员资格：

```
GRANT MEMBERSHIP
IN GROUP personnel
TO M_Haneef;
```

另请参见

- “数据库权限和特权概述”一节第 422 页
- “GRANT 语句”一节 《SQL Anywhere 服务器 - SQL 参考》
- “创建新用户”一节第 430 页

撤消组成员资格

在 Sybase Central 和 Interactive SQL 中您都可以移除组中的用户或组。

将某个用户或组从组中移除不会将它们从数据库（或其它组）中删除。要执行此操作，必须删除用户/组本身。

只有具有 DBA 特权的用户才能撤消组成员资格。

将某个用户添加到某一组后，该用户将继承指派给该组的所有权限。SQL Anywhere 不允许您撤消用户以组成员身份继承的权限的子集，因为您只能撤消由 GRANT 语句显式授予的权限。如果需要让不同的用户拥有不同的权限，则可创建具有适当权限的不同的组，或者为每位用户显式授予他们所需的权限。

◆ 从另一个组中移除用户或组 (Sybase Central):

1. 以具有 DBA 特权的用户身份连接到数据库。
2. 单击 [用户和组]。

3. 双击某一用户或组。
4. 在 [成员资格] 窗格中，右击某一组然后选择 [删除成员资格]。

提示

此外，要删除用户，还可以双击组，然后在右窗格中单击 [成员] 选项卡，再右击用户或组并选择 [删除成员]。

◆ **从另一个组中移除用户或组 (SQL):**

1. 以具有 DBA 特权的用户身份连接到数据库。
2. 执行 REVOKE MEMBERSHIP IN GROUP 语句，指定组名和用户名。

示例

从 personnel 组中移除用户 M_Haneeef:

```
REVOKE MEMBERSHIP  
IN GROUP personnel  
FROM M_Haneeef;
```

另请参见

- “REVOKE 语句”一节 《SQL Anywhere 服务器 - SQL 参考》
- “创建新用户”一节第 430 页
- “从数据库删除用户”一节第 440 页
- “从数据库删除组”一节第 448 页

组的权限和特权

为组授予权限的方法与为任何其它用户 ID 授予权限的方法完全相同。组成员（包括其它组及其成员）会继承针对表、视图和过程的权限。

数据库对象的所有权与单个用户 ID 相关联，不由组成员继承。如果用户 ID personnel 创建了一个表，那么，personnel 用户 ID 就是该表的所有者，其有权更改该表，而且，有权将有关该表的权限授予给其他用户。作为 personnel 的成员的其他用户 ID 不是此表的所有者，也就没有这些权限。只有被授予的权限才会被继承。例如，如果拥有 DBA 特权的用户或 personnel 用户 ID 为 personnel 用户 ID 显式授予针对某个表的 SELECT 权限，则所有的组成员都会继承对该表的 SELECT 访问权限。

也可以将某些特权授予组。

注意

组成员不继承 DBA、RESOURCE 和 GROUP 权限。即使用户 ID 拥有 RESOURCE 特权，personnel 的成员也不会拥有 RESOURCE 特权。

注意

您只能为用户指定 WITH GRANT OPTION。如果将 WITH GRANT OPTION 授予组，则组的成员不会继承 WITH GRANT OPTION。

引用组拥有的表

组可以用于查找数据库中的表和过程。例如，以下查询总会找到视图 SYS.SYSGROUPS，因为所有用户都属于 PUBLIC 组，而 PUBLIC 组又属于拥有 SYSGROUPS 视图的 SYS 组：

```
SELECT * FROM SYSGROUPS;
```

SYSGROUPS 视图包含 *group_name*、*member_name* 对的列表，这些对表示数据库中的组成员资格。

如果名为 employees 的表归用户 ID personnel 所有，且 **M_Haneef** 是 personnel 组的成员，则 **M_Haneef** 可在 SQL 语句中将 employees 表简单地称为 employees。不是 personnel 组成员的用户需要使用限定名称 personnel.employees。

创建一个要拥有表的组

若要允许每个人在访问表时都无需使用限定名称，最好是创建一个专用来拥有这些表的组。不为该组授予任何权限，但让所有用户都成为该组的成员。然后，您可以创建权限组，并在确有必要时为用户授予这些权限组中的成员资格。

如果用户拥有的表与组拥有的表名称相同，则非限定表名将表示用户拥有的表，而非组拥有的表。同样，如果用户属于多个拥有同名表的组，则用户必须限定表名。

请参见“[数据库对象名和前缀](#)”一节第 450 页。

没有口令的组

以组的用户 ID 连接的用户会拥有某些权限。属于组的用户将拥有以该组的用户 ID 名称创建的数据库中的所有表的所有权权限。

可以对数据库进行设置，以便只让 DBA 处理组及其数据库对象，而不允许其它用户 ID 更改组成员资格。可以这样实现这一点：在创建组时，禁止以该组的用户 ID 进行连接。操作方法是：输入 CREATE USER 语句，不带口令。以下语句将创建用户 ID personnel：

```
CREATE USER personnel;
```

该用户 ID 可被授予组权限，而其它用户 ID 则可被授予组成员资格，从而继承已授予 personnel 的任何权限。但是，任何人都不能用 personnel 用户 ID 连接数据库，因为此 ID 没有有效口令。

虽然没有用户可以使用用户 ID personnel 来连接数据库，但是此用户 ID 仍然可以是数据库对象的所有者。CREATE TABLE 语句、CREATE PROCEDURE 语句和 CREATE VIEW 语句都允许将该对象的所有者指定为执行该语句的用户之外的用户。只有具有 DBA 特权的用户才能执行所有权的分配。

特殊组

创建数据库时，还会自动创建 SYS、PUBLIC 和 dbo 组。这些组都没有口令，因此不可以用 SYS、PUBLIC 或 dbo 连接到数据库。但是，这些组都在数据库中起着重要作用。

SYS 组

SYS 组拥有数据库的系统表和视图，它们包含了有关数据库模式的完整说明（包括所有数据库对象和所有用户 ID）。

有关系统表和视图的详细信息，以及有关对表进行访问的说明，请参见“表”《SQL Anywhere 服务器 - SQL 参考》和“系统视图”一节《SQL Anywhere 服务器 - SQL 参考》。

PUBLIC 组

PUBLIC 组拥有对系统表的 SELECT 权限。此外，PUBLIC 组还是 SYS 组的成员，并拥有某些系统表和视图的读取访问权，因此数据库的任何用户都可以了解有关数据库模式的信息。如果要限制此访问权限，可将 PUBLIC 在 SYS 组中的成员资格撤消。

任何新用户 ID 都会自动成为 PUBLIC 组的成员，并继承由拥有 DBA 特权的用户专门授予该组的任何权限。如果需要，您还可以撤消用户在 PUBLIC 中的成员资格。

dbo 组

dbo 组拥有许多系统存储过程和视图。dbo 组是 SYS 组的成员。PUBLIC 组是 dbo 组的成员。dbo 组还拥有用于 UltraLite 和 MobiLink 的表。

从数据库删除组

使用 Sybase Central 和 Interactive SQL 都可以从数据库中删除组。

从数据库中删除用户或组不同于将它们从其它组中移除。从数据库中删除组时，不会从数据库中删除该组的成员，但这些成员会丢失在被删除的组中的成员资格。

只有具有 DBA 特权的用户才能删除组。

◆ 从数据库中删除组 (Sybase Central):

1. 以具有 DBA 特权的用户身份连接到数据库。
2. 单击 [用户和组]。
3. 右击组，然后选择 [删除]。
4. 单击 [是]。

◆ 从数据库中删除组 (SQL):

1. 以具有 DBA 特权的用户身份连接到数据库。
2. 执行 REVOKE CONNECT FROM 语句。

示例

从数据库中移除组 personnel。

```
REVOKE CONNECT FROM personnel;
```

另请参见

- “REVOKE 语句”一节 《SQL Anywhere 服务器 - SQL 参考》
- “撤消用户权限和特权”一节第 439 页
- “从数据库删除用户”一节第 440 页

数据库对象名和前缀

每个数据库对象的名称都是一个标识符。

有关有效标识符规则的信息，请参见“标识符”一节《SQL Anywhere 服务器 - SQL 参考》。

在本手册的所有查询和示例 SQL 语句中，提及示例数据库中的数据库对象时，通常采用的是它们的简单名称。例如：

```
SELECT *
FROM Employees;
```

表、过程和视图都有所有者。DBA 用户 ID 拥有示例数据库中的表。在某些情况下，必须使用所有者用户 ID 作为对象名的前缀，如下语句所示。

```
SELECT *
FROM DBA.Employees;
```

Employees 表引用被称为**限定表引用**。在其它情况下，给出对象名即已足够。本节介绍了何时需要或不需要使用所有者前缀来标识表、视图和过程。

当引用数据库对象时，除以下情况外需要使用前缀：

- 您是数据库对象的所有者。
- 您是拥有数据库对象的组 ID 的成员。

示例

假设以下示例是一个公司的数据库。用户 ID company 创建了所有的表，由于此用户 ID 属于数据库管理员，因此它拥有 DBA 特权。

```
CREATE USER Company
IDENTIFIED BY secret;
GRANT DBA TO Company;
```

company 用户 ID 创建了数据库中的表。

```
CONNECT USER company IDENTIFIED BY secret;
CREATE TABLE company.Customers ( ... );
CREATE TABLE company.Products ( ... );
CREATE TABLE company.Orders ( ... );
CREATE TABLE company.Invoices ( ... );
CREATE TABLE company.Employees ( ... );
CREATE TABLE company.Salaries ( ... );
```

并非公司的每个人都有权访问所有信息。假设销售部门中的两个用户 ID Joe 和 Sally 应该有权访问 Customers、Products 和 Orders 表。要实现这一点，可创建 Sales 组。

```
CREATE USER Sally IDENTIFIED BY xxxxxx;
CREATE USER Joe IDENTIFIED BY xxxxxx;
CREATE USER Sales IDENTIFIED BY xxxxxx;
GRANT GROUP TO Sales;
GRANT ALL ON Customers TO Sales;
GRANT ALL ON Orders TO Sales;
GRANT SELECT ON Products TO Sales;
GRANT MEMBERSHIP IN GROUP Sales TO Sally;
GRANT MEMBERSHIP IN GROUP Sales TO Joe;
```

现在，Joe 和 Sally 有权使用这些表，但他们仍须限定他们的表引用，因为表的所有者是 Company，而 Sally 和 Joe 不是 Company 组的成员：

```
SELECT *  
FROM company.Customers;
```

为了改变这种情况，应使 Sales 组成为 Company 组的成员。

```
GRANT GROUP TO Company;  
GRANT MEMBERSHIP IN GROUP Company TO Sales;
```

现在，Sales 组的成员 Joe 和 Sally 间接地成为了 Company 组的成员，可以引用他们的表而不必使用限定符。现在可以使用以下命令：

```
SELECT *  
FROM Customers;
```

注意

Joe 和 Sally 是 company 组的成员，但他们没有任何额外的权限。company 组没有被明确授予任何表权限。（company 用户 ID 拥有查看诸如 Salaries 之类的表的隐式权限，因为该 ID 创建了该表并具有 DBA 特权。）因此，Joe 和 Sally 在执行以下命令之一时仍会遇到错误：

```
SELECT *  
FROM Salaries;  
SELECT *  
FROM company.Salaries;
```

在这两种情况下，Joe 和 Sally 都没有查看 Salaries 表的权限。

使用视图和过程来获得更高的安全性

对于安全性要求很高的数据库，直接对表定义权限有其局限性。给用户授予的对表的任何权限都会应用于整个表。在许多情况下，需要更精确地授予用户权限，而不仅仅是整个表地授权。例如：

- 不应将员工表中所存储的个人信息或敏感信息的访问权授予那些需要访问该表其它部分的用户。
- 您可能想要授予销售代表更新包含其销售电话记录说明的表的权限，但此权限要仅限于其自己的电话。

在这些情况下，您可以使用视图和存储过程来对权限进行量身定制，以适应贵组织的需要。本节介绍了使用视图和过程对权限进行管理的一些方法。

另请参见

- “使用视图”一节 《SQL Anywhere 服务器 - SQL 的用法》
- “授予针对视图的权限”一节第 435 页

使用视图来定制安全性

视图是计算出的表，其中包含了从基表中选择的行和列。在需要授予用户只对表的一部分进行访问的权限的情况下，视图对于安全非常有用。可以用行或列来定义此部分。例如，您可能想要禁止一组用户查看 `employee` 表的 `Salary` 列，或者可能想要限制用户，仅允许其查看他们已创建的表的行。

示例 1

销售经理需要访问数据库中有关该部门雇员的信息。但是，该经理没有理由访问有关其它部门雇员的信息。

此示例介绍了如何为销售经理创建用户 ID，如何创建可提供其所需信息的视图，以及如何给此销售经理用户 ID 授予相应的权限。

1. 使用 `GRANT` 语句创建新用户 ID。以具有 `DBA` 特权的用户身份登录，然后执行以下语句：

```
CONNECT DBA
IDENTIFIED by sql;

CREATE USER SalesManager
IDENTIFIED BY sales;
```

2. 定义一个只查看销售部门员工的视图，如下所示：

```
CREATE VIEW EmployeeSales AS
SELECT EmployeeID, GivenName, Surname
FROM Employees
WHERE DepartmentID = 200;
```

表引用可由所有者限定，以避免对名称完全相同的表的多义性引用。

3. 授予 `SalesManager` 查看该视图的权限：

```
GRANT SELECT
ON EmployeeSales
TO SalesManager;
```

授予针对视图和针对表的权限时，可以使用完全相同的命令。

示例 2

下一个示例创建一个允许销售经理查看销售订单汇总的视图。定义此视图需要有来自多个表的信息：

1. 创建视图。

```
CREATE VIEW OrderSummary AS
  SELECT OrderDate, Region, SalesRepresentative, CompanyName
  FROM SalesOrders
  KEY JOIN Customers;
```

2. 授予 Sales Manager 查看该视图的权限。

```
GRANT SELECT
ON OrderSummary
TO SalesManager;
```

3. 要检查该过程是否正确执行，请连接到 SalesManager 用户 ID，然后查看您创建的视图：

```
CONNECT SalesManager
IDENTIFIED BY sales;
SELECT *
FROM DBA.EmployeeSales;
SELECT *
FROM DBA.OrderSummary;
```

由于未授予销售经理查看基表的权限，以下命令将产生权限错误。

```
SELECT * FROM GROUPO.Employees;
SELECT * FROM GROUPO.SalesOrders;
```

针对视图的其它权限

前面的示例说明了如何使用视图来定制 SELECT 权限。您可以使用相同的方法授予对视图的 INSERT、DELETE 和 UPDATE 权限。

使用过程来定制安全性

视图可以数据为基础限制访问权限，而过程则限制用户可以执行的操作。用户可以拥有过程的 EXECUTE 权限，而无需对过程所作用的表拥有任何权限。请参见“[授予针对过程的权限](#)”一节第 437 页。

严格的安全性

为获得严格的安全性，您可以禁止对基础表进行的所有访问，并授予用户或用户组执行某些存储过程的权限。此方法严格定义了对数据库中的数据进行修改的方式。

更改嵌套对象的所有权

视图和过程可访问由不同用户拥有的基础对象。例如，如果 `usera`、`userb`、`userc` 和 `userd` 是四个不同的用户，而 `userd.viewd` 可能基于 `userc.viewc`，`userc.viewc` 可能基于 `userb.viewb`，`userb.viewb` 可能基于 `usera.tablea`。对于过程，也与此相似：`userd.procd` 可能调用 `userc.procc`，`userc.procc` 可能调用 `userb.procb`，`userb.procb` 可能插入 `usera.tablea`。

下列 [任意访问控制](DAC) 规则适用于嵌套视图和表：

- 要创建视图，用户必须对视图中的所有基本对象（例如，表和视图）具有 `SELECT` 权限。
- 要访问视图，必须用 `GRANT OPTION`（授权选项）向视图所有者授予对基表或视图的适当权限，还必须向视图的用户授予对视图的适当权限。
- 用 `WHERE` 子句更新需要同时具有 `SELECT` 和 `UPDATE` 权限。
- 如果用户拥有视图定义中的表，则用户可以通过某个视图来访问这些表，即使用户不是该视图的所有者并且尚未被授予对该视图的访问权限，也可以这样做。

下列 DAC 规则适用于嵌套过程：

- 用户无需对基础对象（例如，表、视图或过程）具有任何权限即可创建过程。
- 对于要执行的过程，过程的所有者需要对该过程所引用的对象具有适当的权限。
- 即使某个用户拥有某个过程引用的所有表，除非该用户已被授予对该过程的 `EXECUTE` 权限，否则该用户将无法执行该过程以访问这些表。

下面介绍描述此行为的一些示例。

示例 1：用户 1 创建表 1，用户 2 在表 1 上创建视图 2

- 由于用户 1 是所有者，因此用户 1 总是可以访问表 1。
- 由于用户 1 是基础表的所有者，所以始终可以通过视图 2 访问表 1。即使用户 2 未向用户 1 授予对视图 2 的权限，也同样适用。
- 如果用户 1 向用户 2 授予对表 1 的权限，则用户 2 可直接访问表 1，也可通过视图 2 访问。
- 如果用户 1 向用户 3 授予对表 1 的权限，则用户 3 可访问表 1。
- 如果用户 1 利用授权选项向用户 2 授予对表 1 的权限，*并且*用户 2 向用户 3 授予对视图 2 的权限，则用户 3 可通过视图 2 访问表 1。

示例 2：用户 2 创建访问表 1 的过程 2

- 如果用户 2 向用户 1 授予对过程 2 的 `EXECUTE` 权限，则用户 1 可通过过程 2 访问表 1。注意，这与视图 2 的情形不同，在那种情况下用户 1 不需要具有对视图 2 的权限。

示例 3: 用户 1 创建表 1, 用户 2 创建表 2, 用户 3 创建将表 1 和表 2 连接在一起的视图 3

- 如果用户 1 向用户 3 授予对表 1 的权限, 同时用户 2 向用户 3 授予对表 2 的权限, 则用户 3 可通过视图 3 访问表 1 和表 2。
- 如果用户 3 具有对表 1 的权限而不具有对表 2 的权限, 则用户 3 不能使用视图 3, 即使是访问属于表 1 的列的子集也是如此。
- 如果满足以下条件, 则用户 1 或用户 2 可以使用视图 3: (a) 用户 1 利用授权选项向用户 3 授予对表 1 的权限, (b) 用户 2 利用授权选项向用户 3 授予对表 2 的权限, 且 (c) 用户 3 向用户 1 或用户 2 授予对视图 3 的权限。

如何确定用户权限

组确实使单个用户的权限变得更加复杂。假设用户 `M_Haneef` 分别拥有对特定表的 `SELECT` 和 `UPDATE` 权限，但他同时又是两个组的成员。假设这两个组中的一个组根本没有访问表的权限，另一个只有 `SELECT` 权限。那么，此用户的实际权限是什么呢？

SQL Anywhere 通过以下方式来确定某个用户 ID 是否有执行特定操作的权限：

1. 如果该用户 ID 拥有 `DBA` 特权，则该用户 ID 可以在数据库中执行任何操作。
2. 否则，权限取决于指派给单个用户的权限。如果该用户 ID 已被授予执行某操作的权限，则可以执行此操作。
3. 如果没有为该用户进行过单独设置，则权限取决于该用户所属的各个组的权限。如果这些组中的任何一组有执行某操作的权限，则该用户 ID 将会因其在该组中具有成员资格而拥有相应的权限，并因而可以执行此操作。

此方法最大程度地减少了与权限设置顺序相关的问题。

管理连接所使用的资源

通过构建一个由用户和组所组成的集合，您可管理针对数据库的权限。数据库安全与管理的另一方面是对单个用户可以使用的资源进行限制。

例如，您可能想要阻止单个连接占用过多的可用内存或 CPU 资源，以便能够避免由于某个连接而降低数据库其他用户的连接速度。

SQL Anywhere 提供了具有 DBA 特权的用户可用于控制资源的一组数据库选项。这些选项被称为资源调控器。

设置选项

您可以使用 SET OPTION 语句来设置数据库选项，所用语法如下：

```
SET [ TEMPORARY ] OPTION ... [ userid. | PUBLIC. ]option-name = [ option-value ]
```

可以管理的资源

您可以使用下列选项来管理资源：

- **max_cursor_count** 限制一个连接的游标数量。请参见“[max_cursor_count 选项 \[数据库\]](#)”一节第 517 页。
- **max_statement_count** 为一个连接限制预准备语句的数量。请参见“[max_statement_count 选项 \[数据库\]](#)”一节第 520 页。
- **priority** 设置执行来自连接的请求的优先级级别。请参见“[priority 选项 \[数据库\]](#)”一节第 535 页。
- **max_priority** 控制连接的最大优先级级别。请参见“[max_priority 选项 \[数据库\]](#)”一节第 518 页。

数据库选项设置不能通过组结构继承。

另请参见

- “数据库选项”一节第 466 页
- “SET OPTION 语句”一节 《SQL Anywhere 服务器 - SQL 参考》

目录中的用户和权限

数据库系统视图包含有关数据库当前用户及其权限的信息。

系统视图由特殊用户 ID SYS 所拥有。您不能使用 SYS 用户 ID 进行连接。

拥有 DBA 特权的用户拥有所有系统视图的 SELECT 访问权限，但是对基础系统表却没有这种权限。其他用户对某些表和视图的访问也会受到限制。例如，只有具有 DBA 特权的用户有权访问 SYS.SYSUSERPERM 视图，其中包含有关数据库用户权限的所有信息，以及各个用户 ID 的加密口令。而 SYS.SYSUSERPERMS 是一个包含 SYS.SYSUSERPERM 中除口令以外的所有信息的视图，缺省情况下，所有用户都具有对此视图的 SELECT 访问权限。您可以彻底修改新数据库中为 SYS、PUBLIC、DBA 和 dbo 设置的所有权限和组成员资格。

下表汇总了包含有关用户 ID、组和权限信息的系统视图。用户 ID SYS 拥有列出的所有视图，因此它们的限定名为 SYS.SYSUSERPERM 等。

对这些视图进行的相应 SELECT 查询会生成存储在数据库中的所有用户 ID 信息和权限信息。

视图	缺省	内容
SYSCOLAUTH	PUBLIC	SYSCOLPERM 中的信息，采用了更加易读的格式。请参见“ SYSCOLAUTH 统一视图 ”一节《 SQL Anywhere 服务器 - SQL 参考 》。
SYSCOLPERM	PUBLIC	具有 GRANT 命令所授予的 SELECT 或 UPDATE 权限的所有列。请参见“ SYSCOLPERM 系统视图 ”一节《 SQL Anywhere 服务器 - SQL 参考 》。
DUMMY	PUBLIC	可用于查找当前用户 ID 的虚表。请参见“ DUMMY 系统表 ”一节《 SQL Anywhere 服务器 - SQL 参考 》。
SYSGROUP	PUBLIC	每一个组的每一个成员都由一行代表。请参见“ SYSGROUP 系统视图 ”一节《 SQL Anywhere 服务器 - SQL 参考 》。
SYSGROUPS	PUBLIC	SYSGROUP 中的信息，采用了更加易读的格式。请参见“ SYSGROUPS 统一视图 ”一节《 SQL Anywhere 服务器 - SQL 参考 》。
SYSROCAUTH	PUBLIC	SYSROCPERM 中的信息，采用了更加易读的格式。请参见“ SYSROCAUTH 统一视图 ”一节《 SQL Anywhere 服务器 - SQL 参考 》。
SYSROCPERM	PUBLIC	每行都保存着一个被授予了使用一个过程的权限的用户。请参见“ SYSROCPERM 系统视图 ”一节《 SQL Anywhere 服务器 - SQL 参考 》。

视图	缺省	内容
SYSTABAUTH	PUBLIC	SYSTABLEPERM 中的信息，采用了更加易读的格式。请参见“ SYSTABAUTH 统一视图 ”一节《 SQL Anywhere 服务器 - SQL 参考 》。
SYSTABLEPERM	PUBLIC	由 GRANT 命令授予的对表的所有权限。请参见“ SYSTABLEPERM 系统视图 ”一节《 SQL Anywhere 服务器 - SQL 参考 》。
SYSUSER	仅限 DBA	数据库中所有用户的相关信息。请参见“ SYSUSER 系统视图 ”一节《 SQL Anywhere 服务器 - SQL 参考 》。
SYSUSERAUTH	仅限于 DBA	SYSUSERPERM 中除用户号以外的所有信息。请参见“ SYSUSERAUTH 兼容性视图（不建议使用） ”一节《 SQL Anywhere 服务器 - SQL 参考 》。
SYSUSERAUTHORITY	PUBLIC	授予每个用户 ID 的权限。请参见“ SYSUSERAUTHORITY 系统视图 ”一节《 SQL Anywhere 服务器 - SQL 参考 》。
SYSUSERLIST	PUBLIC	SYSUSERAUTH 中除口令以外的所有信息。请参见“ SYSUSERLIST 兼容性视图（不建议使用） ”一节《 SQL Anywhere 服务器 - SQL 参考 》。
SYSUSERPERM	仅限 DBA	各个用户 ID 的数据库级别的权限和口令。请参见“ SYSUSERPERM 兼容性视图（不建议使用） ”一节《 SQL Anywhere 服务器 - SQL 参考 》。
SYSUSERPERMS	PUBLIC	SYSUSERPERM 中除口令以外的所有信息。请参见“ SYSUSERPERMS 兼容性视图（不建议使用） ”一节《 SQL Anywhere 服务器 - SQL 参考 》。

数据库选项

目录

数据库选项简介	462
---------------	-----

数据库选项简介

数据库选项控制数据库行为的诸多方面。例如，可以将数据库选项用于以下目的：

- **兼容性** 您可以控制您的 SQL Anywhere 数据库操作与 Adaptive Server Enterprise 操作类似的程度，还可控制不符合 SQL/2003 的 SQL 是否会生成错误。
- **错误处理** 您可以控制在出现错误（例如被零除错误或溢出错误）时的操作。
- **并发和事务** 您可以控制并发程度以及 COMMIT 行为的细节。

设置数据库选项

可以使用 SET OPTION 语句设置选项。其语法形式一般如下：

```
SET [ EXISTING ] [ TEMPORARY ] OPTION  
[ userid. | PUBLIC. ]option-name = [ option-value ]
```

如果指定一个用户 ID 或组名，则只为该用户或组设置选项。每个用户都属于 PUBLIC 组。如果未指定用户 ID 或组，则所做的选项更改将应用于发出 SET OPTION 语句的当前登录用户 ID。

任何选项，不论是否是用户定义选项，在可以指派用户特定的值之前都必须具有公共设置。对于用户定义的选项，数据库服务器不支持设置 TEMPORARY 值。

例如，以下语句将选项更改应用于用户 DBA（如果 DBA 是发出该命令的用户）：

```
SET OPTION blocking_timeout = 3;
```

以下语句会将更改应用于 PUBLIC 用户 ID（所有用户都属于该 ID 代表的用户组）。要执行此语句，必须具有 DBA 权限。

```
SET OPTION PUBLIC.login_mode = 'Standard';
```

如果忽略 *option-value*，将从数据库中删除指定的选项设置。如果它是一个个人选项设置，则它的值会恢复为 PUBLIC 设置。如果删除一个 TEMPORARY 选项，则选项设置会恢复为永久设置。

请参见“SET OPTION 语句”一节《SQL Anywhere 服务器 - SQL 参考》。

◆ 设置数据库的选项 (Sybase Central)

1. 打开数据库服务器。
2. 右击数据库并选择 [选项]。
3. 编辑值。

提示

您还可以使用 [数据库选项] 窗口为特定的用户和组设置数据库选项（为用户或组打开此窗口时，此窗口分别叫作 [用户选项] 窗口或 [组选项] 窗口）。

当您为数据库本身设置选项时，您实际是在为该数据库中的 PUBLIC 组设置选项，这是因为所有用户和组都从 PUBLIC 继承选项设置。

小心

不支持在从游标读取行时更改选项设置，因为这会导致不可靠的结果。例如，在从游标读取时更改 `date_format` 设置会导致在结果集的行中出现不同的日期格式。不要在读取行时更改选项设置。

注意

在使用土耳其语归类或区分大小写的数据库中，如果选项名称使用错误的大小写，在 `SYSOPTION` 上执行查询或执行类似以下的查询都可能匹配不到任何行：

```
SELECT * FROM sa_conn_properties( ) WHERE propname = 'BLOCKING';
```

有关选项名称的正确大小写的信息，请参见“[按字母顺序排列的选项列表](#)”一节第 477 页。

数据库选项的作用域和持续时间

可以将选项设置为三种级别的作用域：公共、用户和临时。

临时选项优先于用户设置和公共设置。用户级选项优先于公共设置。如果为当前用户设置用户级选项，则相应的临时选项也会被设置。

某些选项（例如 `COMMIT` 行为）的作用域是数据库范围。设置这些选项要求具有 `DBA` 权限。其它选项（例如 `isolation_level`）也可以只用于当前连接，并且不需要特殊权限。

根据选项的不同，选项设置更改的时间也不同。对全局选项（如 `recovery_time`）的更改发生在下一次数据库启动时。

通常，只有影响到当前连接的选项才会立即生效。例如，可以在一个事务中间更改选项设置。该规则的一个例外是 *在游标打开时更改选项会导致不可靠的结果*。例如，在游标打开时，更改 `date_format` 或许不会改变下一行的格式。根据游标检索的方式，或许要到几行之后更改才会对用户生效。

设置公共选项

为 `PUBLIC` 用户 ID 设置选项要求具有 `DBA` 权限。

为 `PUBLIC` 用户 ID 更改选项的值，相当于为没有设置该值的所有用户设置此选项的永久值。如果某选项没有 `PUBLIC` 用户 ID 设置，则无法为单个用户 ID 设置此选项的值。

一些只能为 `PUBLIC` 用户设置的选项对现有连接会立即生效，即使用户通过 `CONNECTION_PROPERTY` 函数无法看到更改的设置也是这样。`global_database_id` 选项就是一例。因此，在有其他用户连接到数据库时，不应更改仅用于 `PUBLIC` 的选项。

设置临时选项

向 `SET OPTION` 语句添加 `TEMPORARY` 关键字可改变此项更改的持续时间。通常选项更改是永久的。在使用 `SET OPTION` 语句显式进行更改之前，它不会变化。

执行 `SET TEMPORARY OPTION` 语句时，新选项值只对当前连接生效，其有效时间也只是连接的持续时间。

在使用 SET TEMPORARY OPTION 设置 PUBLIC 选项时，更改在数据库运行的任何时间都有效。当数据库关闭时，PUBLIC 用户 ID 的临时选项恢复为其永久值。

为 PUBLIC 用户 ID 设置临时选项在安全上具有优点。例如，在启用 login_mode 选项时，数据库依赖其运行时所在的系统的登录安全性。将其作为临时选项设置启用意味着，对于依赖 Windows 域的安全性的数据库，如果该数据库关闭并被复制到本地计算机，它的安全不会受到威胁。在这种情况下，login_mode 选项将恢复至其永久值，而其永久值可能为 [Standard]，即不允许集成登录的模式。

SQL 语句的设置选项

INSERT、UPDATE、DELETE、SELECT、UNION、EXCEPT 和 INTERSECT 语句有一个 OPTION 子句，用于指定语句使用实例化视图的方式和查询优化方式。此子句也可用于指定一个仅适用于该语句的选项设置，该选项设置优先于任何有效的公共或临时选项设置。在 OPTION 子句中，可以更改以下选项的设置：

- isolation_level
- max_query_tasks
- optimization_goal
- optimization_level
- optimization_workload
- user_estimates

查找选项设置

可以使用多种方法获得选项设置的列表或单个选项的值。

获取选项值的列表

- 您的连接的当前选项设置作为**连接属性**的一个子集提供。您可以使用 sa_conn_properties 系统过程列出所有连接属性。

```
CALL sa_conn_properties;
```

要按字母顺序排序此列表，可以执行以下语句：

```
SELECT *  
FROM sa_conn_properties( )  
ORDER BY PropName;
```

如果希望过滤结果或对结果按名称以外的项目进行排序，还可使用 WHERE 子句。例如：

```
SELECT *  
FROM sa_conn_properties( )  
WHERE PropDescription LIKE '%cache%'  
ORDER BY PropNum;
```

请参见“sa_conn_properties 系统过程”一节《SQL Anywhere 服务器 - SQL 参考》。

- 在 Interactive SQL 中，不带参数的 SET 语句可列出选项的当前设置。

```
SET;
```

- 在 Sybase Central 中，选择数据库，然后选择 [文件] » [选项]。
- 在 SYSOPTIONS 系统视图上使用以下查询，以显示所有 PUBLIC 值以及已显式设置的 USER 值：

```
SELECT *  
FROM SYSOPTIONS;
```

获取单个选项值

可以使用 CONNECTION_PROPERTY 系统函数获得单个设置。例如，以下语句会报告 ansi_blanks 选项的值：

```
SELECT CONNECTION_PROPERTY ( 'ansi_blanks' );
```

请参见“CONNECTION_PROPERTY 函数 [System]”一节《SQL Anywhere 服务器 - SQL 参考》。

监控选项设置

可以在尝试设置数据库选项时，使用 sa_server_option 系统过程指示数据库服务器发送消息或返回错误。

可使用 OptionWatchList 属性创建想要监控的选项列表，使用 OptionWatchAction 属性指定在尝试设置要监控的选项时数据库服务器要执行的操作。

例如，以下命令指示数据库服务器监控数据库选项 automatic_timestamp、float_as_double 和 tsq_hex_constant：

```
CALL dbo.sa_server_option (  
    'OptionWatchList','automatic_timestamp,float_as_double,tsq_hex_constant' );
```

如果尝试设置在 OptionWatchList 属性中指定的选项，以下命令指示数据库服务器返回错误：

```
CALL dbo.sa_server_option( 'OptionWatchAction','ERROR' );
```

另请参见

- “sa_server_option 系统过程”一节《SQL Anywhere 服务器 - SQL 参考》
- OptionWatchAction 和 OptionWatchList 属性：“数据库服务器属性”一节第 588 页

初始选项设置

可以通过 TDS 协议（Open Client 和 jConnect JDBC 连接）或通过 SQL Anywhere 协议（ODBC 和嵌入式 SQL）连接到 SQL Anywhere。

如果您的用户既使用 TDS 又使用 SQL Anywhere 特定的协议，您可以使用存储过程来配置它们的初始设置。SQL Anywhere 使用此方法设置 Open Client 连接和 jConnect 连接，以反映 Adaptive Server Enterprise 的缺省行为。

初始设置使用 login_procedure 选项进行控制。此选项命名了一个存储过程以在用户连接时使用。缺省设置为使用 sp_login_environment 系统过程。需要时您可以更改此行为。

然后，sp_login_environment 检查是否通过 TDS 进行连接。如果是，则它调用 sp_tsq_environment 过程，该过程会将当前连接的几个选项设置为新的缺省值。

另请参见

- “login_procedure 选项 [数据库]” 一节第 513 页
- “sp_login_environment 系统过程” 一节 《SQL Anywhere 服务器 - SQL 参考》
- “sp_tsq_environment 系统过程” 一节 《SQL Anywhere 服务器 - SQL 参考》

删除选项设置

如果忽略 *option-value*，将从数据库中删除指定的选项设置。如果它是一个个人选项设置，则它的值会恢复为 PUBLIC 设置。如果删除一个 TEMPORARY 选项，则选项设置会恢复为永久设置。

例如，以下语句将 ansi_blanks 选项重置为其缺省值：

```
SET OPTION ansi_blanks =;
```

请参见 “SET OPTION 语句” 一节 《SQL Anywhere 服务器 - SQL 参考》。

选项分类

SQL Anywhere 提供了许多选项。将这些选项分成几个大类会方便一些。选项的分类为：

- “数据库选项” 一节第 466 页
- “兼容性选项” 一节第 472 页
- “SQL Remote 选项” 一节第 476 页
- “Interactive SQL 选项” 一节第 667 页

数据库选项

本节列出所有数据库选项。

选项	值	缺省
“allow_snapshot_isolation 选项 [数据库]” 一节第 478 页	On、Off	Off
“auditing 选项 [数据库]” 一节第 484 页	On、Off	Off
“auditing_options 选项 [数据库]” 一节第 485 页	保留	保留
“background_priority 选项 [数据库] [不建议使用]” 一节第 485 页	On、Off	Off
“blocking 选项 [数据库]” 一节第 486 页	On、Off	On

选项	值	缺省
“blocking_timeout 选项 [数据库]” 一节第 487 页	整数（以毫秒为单位）	0
“checkpoint_time 选项 [数据库]” 一节第 488 页	分钟数	60
“cis_option 选项 [数据库]” 一节第 488 页	0, 7	0
“cis_rowset_size 选项 [数据库]” 一节第 489 页	整数	50
“collect_statistics_on_dml_updates 选项 [数据库]” 一节第 490 页	On、Off	On
“conn_auditing 选项 [数据库]” 一节第 491 页	On、Off	On
“connection_authentication 选项 [数据库]” 一节第 491 页	字符串	空字符串
“cooperative_commit_timeout 选项 [数据库]” 一节第 493 页	整数（以毫秒为单位）	250
“cooperative_commits 选项 [数据库]” 一节第 494 页	On、Off	On
“database_authentication [数据库]” 一节第 495 页	字符串	空字符串
“date_format 选项 [数据库]” 一节第 496 页	字符串	YYYY-MM-DD
“date_order 选项 [数据库]” 一节第 497 页	MDY、YMD、DMY	YMD
“debug_messages 选项 [数据库]” 一节第 498 页	On、Off	Off
“dedicated_task 选项 [数据库]” 一节第 499 页	On、Off	Off
“default_dbpace 选项 [数据库]” 一节第 499 页	字符串	空字符串（使用系统 dbpace）
“default_timestamp_increment 选项 [数据库] [MobiLink 客户端]” 一节第 500 页	整数（1 到 1000000，以微秒为单位）	1

选项	值	缺省
“ delayed_commit_timeout 选项 [数据库]” 一节 第 501 页	整数（以毫秒为单位）	500
“ delayed_commits 选项 [数据库]” 一节 第 501 页	On、Off	Off
“ exclude_operators 选项 [数据库]” 一节 第 503 页	保留	保留
“ extended_join_syntax 选项 [数据库]” 一节 第 503 页	On、Off	On
“ first_day_of_week 选项 [数据库]” 一节 第 504 页	1, 2, 3, 4, 5, 6, 7	7（星期日是一周中的第一天）
“ for_xml_null_treatment 选项 [数据库]” 一节 第 505 页	Empty、Omit	Omit
“ force_view_creation 选项 [数据库]” 一节 第 506 页	保留	保留
“ global_database_id 选项 [数据库]” 一节 第 506 页	整数	2147483647
“ http_session_timeout 选项 [数据库]” 一节 第 507 页	整数	30
“ integrated_server_name 选项 [数据库]” 一节 第 507 页	字符串	NULL
“ isolation_level 选项 [数据库] [兼容性]” 一节 第 508 页	0、1、2、3、快照、语句快照、只读语句快照	0
“ java_location 选项 [数据库]” 一节 第 509 页	字符串	空字符串
“ java_main_userid 选项 [数据库]” 一节 第 510 页	字符串	缺省 DBA 用户
“ java_vm_options 选项 [数据库]” 一节 第 510 页	字符串	空字符串
“ log_deadlocks 选项 [数据库]” 一节 第 511 页	On、Off	Off

选项	值	缺省
“login_mode 选项 [数据库]” 一节第 512 页	Standard、Integrated、Kerberos、Mixed (不建议使用)	Standard
“login_procedure 选项 [数据库]” 一节第 513 页	字符串	sp_login_environment
“materialized_view_optimization 选项 [数据库]” 一节第 515 页	Disabled、Fresh、Stale、N { Minute[s] Hour[s] Day[s] Week[s] Month[s] }	Stale
“max_client_statements_cached 选项 [数据库]” 一节第 516 页	整数	10
“max_cursor_count 选项 [数据库]” 一节第 517 页	整数	50
“max_plans_cached 选项 [数据库]” 一节第 517 页	整数	20
“max_priority 选项 [数据库]” 一节第 518 页	Critical、High、Above Normal、Normal、Below normal、Low、Background	Normal
“max_query_tasks 选项 [数据库]” 一节第 519 页	整数	0
“max_recursive_iterations 选项 [数据库]” 一节第 520 页	整数	100
“max_statement_count 选项 [数据库]” 一节第 520 页	大于等于 0 的整数	50
“max_temp_space 选项 [数据库]” 一节第 521 页	整数 [k m g p]	0
“min_password_length 选项 [数据库]” 一节第 522 页	大于等于 0 的整数	0 个字符

选项	值	缺省
“odbc_describe_binary_as_varbinary [数据库]” 一节第 524 页	On、Off	Off
“odbc_distinguish_char_and_varchar 选项 [数据库]” 一节第 525 页	On、Off	Off
“oem_string 选项 [数据库]” 一节第 525 页	字符串（最多 128 字节）	空字符串
“on_charset_conversion_failure 选项 [数据库]” 一节第 527 页	Ignore、Warning、Error	Ignore
“optimization_goal 选项 [数据库]” 一节第 528 页	First-row 或 All-rows	All-rows
“optimization_level 选项 [数据库]” 一节第 529 页	0-15	9
“optimization_workload 选项 [数据库]” 一节第 530 页	Mixed、OLAP	Mixed
“pinned_cursor_percent_of_cache 选项 [数据库]” 一节第 531 页	0-100 之间的整数	10
“post_login_procedure 选项 [数据库]” 一节第 531 页	字符串	空字符串
“precision 选项 [数据库]” 一节第 533 页	1 到 127（含）之间的整数	30
“prefetch 选项 [数据库]” 一节第 534 页	Off、Conditional、Always	Conditional
“preserve_source_format 选项 [数据库]” 一节第 534 页	On、Off	On
“prevent_article_pkey_update 选项 [数据库] [MobiLink 客户端]” 一节第 535 页	On、Off	On
“priority 选项 [数据库]” 一节第 535 页	Critical、High、Above Normal、Normal、Below normal、Low、Background	Normal

选项	值	缺省
“quoted_identifier 选项 [兼容性]” 一节 第 537 页	On、Off	On
“read_past_deleted 选项 [数据库]” 一节 第 538 页	On、Off	On
“recovery_time 选项 [数据库]” 一节第 538 页	整数（以分钟为单位）	2
“remote_idle_timeout 选项 [数据库]” 一节 第 539 页	整数（以秒为单位）	15
“request_timeout 选项 [数据库]” 一节 第 540 页	整数（0 到 86400， 以秒为单位）	0
“return_date_time_as_string 选项 [数据库]” 一节 第 541 页	On、Off	Off
“rollback_on_deadlock [数据库]” 一节 第 542 页	On、Off	On
“row_counts 选项 [数据库]” 一节第 542 页	On、Off	Off
“scale 选项 [数据库]” 一节第 543 页	0 到 127（含）之 间的整数，小于 为 precision 数据库 选项指定的值	6
“secure_feature_key [数据库]” 一节第 544 页	字符串	NULL
“sort_collation 选项 [数据库]” 一节第 544 页	Internal、 collation_name 或 collation_id	Internal
“subsume_row_locks 选项 [数据库]” 一节 第 550 页	On、Off	On
“suppress_tds_debugging 选项 [数据库]” 一节 第 551 页	On、Off	Off
“synchronize_mirror_on_commit 选项 [数据 库]” 一节第 551 页	On、Off	Off
“tds_empty_string_is_null 选项 [数据库]” 一节 第 552 页	On、Off	Off

选项	值	缺省
“temp_space_limit_check 选项 [数据库]” 一节第 552 页	On、Off	On
“time_zone_adjustment 选项 [数据库]” 一节第 554 页	放在引号中的整数或负整数，或者是放在引号中的字符串（以小时和分钟表示时间，在 + 或 - 之后）	根据客户端的连接类型，由客户端或数据库服务器的时区进行设置
“truncate_timestamp_values 选项 [数据库] [MobiLink 客户端]” 一节第 556 页	On、Off	Off
“updatable_statement_isolation 选项 [数据库]” 一节第 557 页	0, 1, 2, 3	0
“update_statistics 选项 [数据库]” 一节第 558 页	On、Off	On
“user_estimates 选项 [数据库]” 一节第 559 页	Enabled、Disabled、Override-Magic	Override-Magic
“verify_password_function 选项 [数据库]” 一节第 560 页	字符串	空字符串
“wait_for_commit 选项 [数据库]” 一节第 564 页	On、Off	Off
“webservice_namespace_host 选项 [数据库]” 一节第 564 页	NULL、hostname-string	NULL

兼容性选项

以下的选项允许 SQL Anywhere 行为与 Adaptive Server Enterprise 行为兼容，或者既支持旧行为，也允许 ISO SQL/2003 行为。

为实现与 Adaptive Server Enterprise 的进一步兼容，可使用 Transact-SQL SET 语句替代 SQL Anywhere SET OPTION 语句在当前连接期间设置部分选项。请参见“[SET 语句 \[T-SQL\]](#)”一节《[SQL Anywhere 服务器 - SQL 参考](#)》。

缺省设置

这些选项中，有一些的缺省设置与 Adaptive Server Enterprise 的缺省设置不同。为确保您的 SQL Anywhere 和 Adaptive Server Enterprise 数据库的兼容性，应显式设置本节列出的每个兼容性选项。

在使用 Open Client 或 JDBC 接口进行连接时，为使当前连接与 Adaptive Server Enterprise 兼容，某些选项设置将被显式设置。下表列出了这些选项。

与 Adaptive Server Enterprise 兼容的 Open Client 和 JDBC 连接选项

选项	设置
allow_nulls_by_default	Off
ansi_blanks	Off
ansi_substring	On
ansinull	On
chained	Off
continue_after_raiseerror	On
escape_character	Off
on_tsq_l_error	对 JConnect 连接，继续执行
time_format	HH:NN:SS.SSS
timestamp_format	YYYY-MM-DD HH:NN:SS.SSS
tsq_l_outer_joins	Off
tsq_l_variables	On

Transact-SQL 和 SQL/2003 的兼容性选项

下表列出了兼容性选项、其允许值及缺省设置。

选项	值	缺省值
“allow_nulls_by_default 选项 [兼容性]” 一节第 477 页	On、Off	On
“ansi_blanks 选项 [兼容性]” 一节第 480 页	On、Off	Off
“ansi_close_cursors_on_rollback 选项 [兼容性]” 一节第 481 页	On、Off	Off
“ansi_permissions 选项 [兼容性]” 一节第 481 页	On、Off	On

选项	值	缺省值
“ansi_update_constraints 选项 [兼容性]” 一节第 483 页	Off、Cursors、Strict	Cursors
“ansinull 选项 [兼容性]” 一节第 483 页	On、Off	On
“chained 选项 [兼容性]” 一节第 487 页	On、Off	On
“close_on_endtrans 选项 [兼容性]” 一节第 489 页	On、Off	On
“continue_after_raiseerror 选项 [兼容性]” 一节第 492 页	On、Off	On
“conversion_error 选项 [兼容性]” 一节第 493 页	On、Off	On
“date_format 选项 [数据库]” 一节第 496 页	字符串	YYYY-MM-DD
“date_order 选项 [数据库]” 一节第 497 页	MDY、YMD、DMY	YMD
“escape_character 选项 [兼容性]” 一节第 502 页	保留	保留
“fire_triggers 选项 [兼容性]” 一节第 504 页	On、Off	On
“isolation_level 选项 [数据库][兼容性]” 一节第 508 页	0, 1, 2, 3	0
“nearest_century 选项 [兼容性]” 一节第 523 页	0 到 100 (含) 之间的整数	50
“non_keywords 选项 [兼容性]” 一节第 523 页	字符串 (以逗号分隔的关键字列表)	空字符串 (不关闭任何关键字)
“on_tsq_error 选项 [兼容性]” 一节第 528 页	Stop、Conditional、Continue	Conditional
“quoted_identifier 选项 [兼容性]” 一节第 537 页	On、Off	On

选项	值	缺省值
“sql_flagger_error_level 选项 [兼容性]” 一节第 545 页	Off、SQL:1992/Entry、SQL:1992/Intermediate、SQL:1992/Full、SQL:1999/Core、SQL:1999/Package、SQL:2003/Core、SQL:2003/Package、Ultralite	Off
“sql_flagger_warning_level 选项 [兼容性]” 一节第 546 页	Off、SQL:1992/Entry、SQL:1992/Intermediate、SQL:1992/Full、SQL:1999/Core、SQL:1999/Package、SQL:2003/Core、SQL:2003/Package、Ultralite	Off
“string_truncation 选项 [兼容性]” 一节第 549 页	On、Off	On
“time_format 选项 [兼容性]” 一节第 553 页	字符串	HH:NN:SS.SSS
“timestamp_format 选项 [兼容性]” 一节第 554 页	字符串	YYYY-MM-DD HH:NN:SS.SSS
“tsql_outer_joins 选项 [兼容性]” 一节第 557 页	On、Off	Off
“tsql_variables 选项 [兼容性]” 一节第 557 页	On、Off	Off

同步选项

可设置以下数据库选项，以配置作为 MobiLink 同步客户端使用的 SQL Anywhere 数据库。

选项	值	缺省值
“default_timestamp_increment 选项 [数据库] [MobiLink 客户端]” 一节第 500 页	整数（1 到 1000000，以微秒为单位）	1
“delete_old_logs 选项 [MobiLink 客户端] [SQL Remote] [复制代理]” 一节第 502 页	On、Off、Delay、 <i>n</i> days	Off
“prevent_article_pkey_update 选项 [数据库] [MobiLink 客户端]” 一节第 535 页	On、Off	On
“truncate_timestamp_values 选项 [数据库] [MobiLink 客户端]” 一节第 556 页	On、Off	Off

SQL Remote 选项

包含以下选项以提供对 SQL Remote 复制行为的控制。

选项	值	缺省值
“blob_threshold 选项 [SQL Remote]” 一节第 486 页	整数（以字节为单位）	256
“compression 选项 [SQL Remote]” 一节第 490 页	从 -1 到 9 之间的整数	6
“delete_old_logs 选项 [MobiLink 客户端] [SQL Remote] [复制代理]” 一节第 502 页	On、Off、Delay、 <i>n</i> days	Off
“external_remote_options [SQL Remote]” 一节第 503 页	On、Off	Off
“qualify_owners 选项 [SQL Remote]” 一节第 536 页	On、Off	On
“quote_all_identifiers 选项 [SQL Remote]” 一节第 537 页	On、Off	Off
“replication_error 选项 [SQL Remote]” 一节第 539 页	存储过程名	（无过程）
“replication_error_piece 选项 [SQL Remote]” 一节第 540 页	存储过程名	（无过程）
“save_remote_passwords 选项 [SQL Remote]” 一节第 543 页	On、Off	On

选项	值	缺省值
“sr_date_format 选项 [SQL Remote]” 一节第 547 页	<i>date-string</i>	<i>YYYY/MM/DD</i>
“sr_time_format 选项 [SQL Remote]” 一节第 548 页	<i>time-string</i>	<i>HH:NN:SS.SSSSSS</i>
“sr_timestamp_format [SQL Remote]” 一节第 549 页	<i>timestamp-string</i>	<i>YYYY/MM/DD HH:NN:SS.SSSSSS</i>
“subscribe_by_remote 选项 [SQL Remote]” 一节第 550 页	On、Off	On
“verify_all_columns 选项 [SQL Remote]” 一节第 560 页	On、Off	Off
“verify_threshold 选项 [SQL Remote]” 一节第 563 页	整数（以字节为单位）	1000

复制代理选项

包含以下选项以提供对复制代理复制行为的控制。

选项	值	缺省值
“delete_old_logs 选项 [MobiLink 客户端] [SQL Remote] [复制代理]” 一节第 502 页	On、Off、Delay、 <i>n</i> days	Off
“replicate_all 选项 [复制代理]” 一节第 539 页	On、Off	Off

按字母顺序排列的选项列表

本节按照字母顺序列出选项。

allow_nulls_by_default 选项 [兼容性]

控制是否允许创建时未指定 NULL 或 NOT NULL 的新列包含 NULL 值。

允许值

On、Off

缺省值

On

对 Open Client 和 jConnect 连接为 Off

注释

allow_nulls_by_default 选项是为与 Transact-SQL 兼容而加入的。请参见“为实现 Transact-SQL 兼容性设置选项”一节《SQL Anywhere 服务器 - SQL 的用法》。

allow_read_client_file 选项 [数据库]

控制是否允许读取客户端计算机上的文件。

允许值

On、Off

缺省值

Off

作用域

需要 DBA 权限。

注释

必须启用此选项才能读取客户端计算机上的文件，例如，使用 EAD_CLIENT_FILE 函数。

另请参见

- “访问客户端计算机上的数据”一节《SQL Anywhere 服务器 - SQL 的用法》
- “READ_CLIENT_FILE 函数 [String]”一节《SQL Anywhere 服务器 - SQL 参考》
- “READCLIENTFILE 特权”一节第 426 页
- “LOAD TABLE 语句”一节《SQL Anywhere 服务器 - SQL 参考》
- “isql_allow_read_client_file 选项 [Interactive SQL]”一节第 673 页
- “allow_write_client_file 选项 [数据库]”一节第 479 页
- “isql_allow_write_client_file 选项 [Interactive SQL]”一节第 674 页
- “客户端数据安全性”一节《SQL Anywhere 服务器 - SQL 的用法》

allow_snapshot_isolation 选项 [数据库]

控制是启用还是禁用快照隔离。

允许值

On、Off

缺省值

Off

作用域

只能为 PUBLIC 组设置。需要 DBA 权限。

注释

此选项控制是否为数据库启用快照隔离。一旦此选项设置为 On，则事务使用快照隔离时，数据库服务器开始在临时文件中记录已更新行的初始版本。

如果在更改 `allow_snapshot_isolation` 选项的设置时有事务在进行，则更改不立即生效。当选项设置从 Off 更改为 On 时，如果有事务正在运行，则必须完成这些事务才能使用快照。当选项设置从 On 更改为 Off 时，允许在数据库服务器停止收集版本信息之前完成任何未完成的快照，并且不启动新快照。

可以通过查询 `SnapshotIsolationState` 数据库属性的值查看数据库的当前快照隔离设置：

```
SELECT DB_PROPERTY ( 'SnapshotIsolationState' );
```

`SnapshotIsolationState` 属性可为以下值之一：

- **On** 为数据库启用快照隔离。
- **Off** 为数据库禁用快照隔离。
- **in_transition_to_on** 一旦当前事务完成，就将启用快照隔离。
- **in_transition_to_off** 一旦当前事务完成，就将禁用快照隔离。

另请参见

- [“isolation_level 选项 \[数据库\] \[兼容性\]”一节第 508 页](#)
- [“updatable_statement_isolation 选项 \[数据库\]”一节第 557 页](#)
- [“快照隔离”一节 《SQL Anywhere 服务器 - SQL 的用法》](#)
- [“隔离级别和一致性”一节 《SQL Anywhere 服务器 - SQL 的用法》](#)
- [“启用快照隔离”一节 《SQL Anywhere 服务器 - SQL 的用法》](#)

示例

以下语句为数据库启用快照隔离：

```
SET OPTION PUBLIC.allow_snapshot_isolation = 'On';
```

`allow_write_client_file` 选项 [数据库]

控制是否允许向客户端计算机写入文件。

允许值

On、Off

缺省值

Off

作用域

需要 DBA 权限。

注释

必须启用此选项才能向客户端计算机写入文件，例如，使用 WRITE_CLIENT_FILE 函数。

另请参见

- “访问客户端计算机上的数据”一节 《SQL Anywhere 服务器 - SQL 的用法》
- “WRITE_CLIENT_FILE 函数 [String]”一节 《SQL Anywhere 服务器 - SQL 参考》
- “WRITECLIENTFILE 特权”一节第 427 页
- “UNLOAD 语句”一节 《SQL Anywhere 服务器 - SQL 参考》
- “isql_allow_write_client_file 选项 [Interactive SQL]”一节第 674 页
- “allow_read_client_file 选项 [数据库]”一节第 478 页
- “isql_allow_read_client_file 选项 [Interactive SQL]”一节第 673 页
- “客户端数据安全性”一节 《SQL Anywhere 服务器 - SQL 的用法》

ansi_blanks 选项 [兼容性]

控制字符数据在客户端截断时的行为。

允许值

On、Off

缺省值

Off

注释

只有在数据库忽略字符串比较中的尾随空白，并填补读到字符数组的字符串时，ansi_blanks 选项才有效。在将数据类型为 CHAR(N) 的值读入一个 C char[M] 变量并且 N 大于或等于 M 时，它会强制出现截断错误。若将 ansi_blanks 选项设置为 Off，则只有在截断至少一个非空字符时才出现截断错误。

对于嵌入式 SQL，如果 ansi_blanks 选项设置为 On，则在提供数据类型为 DT_STRING 的值时，必须将 sqllen 字段设置为包含该值的缓冲区的长度（至少为该值的长度加上终止空字符的空间）。若 ansi_blanks 选项设置为 Off，则该长度只由空字符的位置决定。ansi_blanks 选项的值在建立连接时确定。建立连接后，更改该选项将不影响此 sqllen 嵌入式 SQL 行为。

若数据库以空白填补，则如果正在读取的表达式是 CHAR 或 NCHAR（不是 VARCHAR 或 NVARCHAR）并且正在被读取到 char 或 nchar（不是 VARCHAR 或 NVARCHAR）主机变量，则该选项控制发送到客户端的截断警告。如果满足这些条件，并且主机变量太小，不足以在读取的表达式以空白形式填补最大长度时容纳该表达式，则会引发截断警告，并且指示符中包含容纳读取的表达式所需的最小字节数（如果该表达式是用空白填补其最大长度）。如果表达式是 CHAR(N)

或 NCHAR(N)，则指示符也可能被设置为 N 以外的其它值，以反映返回值的字符集转换结果和由于字符长度的不同造成的影响。

ansi_close_cursors_on_rollback 选项 [兼容性]

控制执行 ROLLBACK 时是否关闭使用 WITH HOLD 打开的游标。

允许值

On、Off

缺省值

Off

注释

SQL/3 标准草案要求在事务回退时关闭所有游标。缺省情况下，SQL Anywhere 在回退过程中只关闭那些没有使用 WITH HOLD 子句打开的游标。此选项允许您强制关闭所有游标。

close_on_endtrans 选项会替代 ansi_close_cursors_on_rollback 选项。

另请参见

- [“close_on_endtrans 选项 \[兼容性\]”一节第 489 页](#)

ansi_permissions 选项 [兼容性]

为 DELETE 和 UPDATE 语句控制权限检查。

允许值

On、Off

缺省值

On

作用域

只能为 PUBLIC 组设置。设置立即生效。需要 DBA 权限。

注释

若 ansi_permissions 选项设置为 On，则检查 DELETE 和 UPDATE 语句的 SQL/2003 权限要求。在 Adaptive Server Enterprise 中缺省值为 Off。下表简要介绍了其中的差别。

SQL 语句	ansi_permissions 关闭时所需的权限	ansi_permissions 启用时所需的权限
UPDATE	正在对其设置值的列上的 UPDATE 权限	正在对其设置值的列上的 UPDATE 权限 出现在 WHERE 子句中的所有列上的 SELECT 权限 SET 子句右侧的所有列上的 SELECT 权限
DELETE	表上的 DELETE 权限	表上的 DELETE 权限 出现在 WHERE 子句中的所有列上的 SELECT 权限

只能为 PUBLIC 组设置 ansi_permissions 选项。不允许专用设置。

ansi_substring 选项 [兼容性]

控制 SUBSTRING (SUBSTR) 函数在为启动参数或长度参数提供了负值时的行为。

允许值

Off、On

缺省值

On

作用域

可以为单个连接或 PUBLIC 组设置此选项。设置立即生效。

说明

ansi_substring 选项设置为 On 时，SUBSTRING 函数的行为相当于 ANSI/ISO SQL/2003 行为。如果起始偏移为负或零，则视为字符串左侧用非字符填补，并且在提供负值长度的情况下会出现错误。

当此选项设置为 Off 时，SUBSTRING 函数的行为与先前版本的 SQL Anywhere 的行为相同：负的起始偏移表示从字符串末尾开始的偏移，而负的长度表示所需的子串在起始偏移左侧第 N 个字符处结束，其中 N 为长度。并且，起始偏移 0 等效于起始偏移 1。

此选项的设置并不影响 BYTE_SUBSTR 函数的行为。建议在使用 SUBSTRING 函数时避免使用非正数的起始偏移或负长度。请尽可能改用 LEFT 或 RIGHT 函数。

另请参见

- “SUBSTRING 函数 [String]” 一节 《SQL Anywhere 服务器 - SQL 参考》
- “LEFT 函数 [String]” 一节 《SQL Anywhere 服务器 - SQL 参考》
- “RIGHT 函数 [String]” 一节 《SQL Anywhere 服务器 - SQL 参考》

示例

以下示例说明了基于不同 `ansi_substring` 选项设置的不同 `SUBSTRING` 函数返回值。

```
SUBSTRING( 'abcdefgh',-2,4 );
ansi_substring = Off ==> 'gh' // substring starts at second-last character
ansi_substring = On  ==> 'a'  // takes the first 4 characters of
                             // ???abcdefgh and discards all ?

SUBSTRING( 'abcdefgh',4,-2 );
ansi_substring = Off ==> 'cd'
ansi_substring = On  ==> value -2 out of range for destination

SUBSTRING( 'abcdefgh',0,4 );
ansi_substring = Off ==> 'abcd'
ansi_substring = On  ==> 'abc'
```

ansi_update_constraints 选项 [兼容性]

控制允许的更新的范围。

允许值

Off、Cursors、Strict

缺省值

Cursors

注释

SQL Anywhere 提供了几种扩展，它们允许进行 ANSI SQL 标准所不允许的更新。这些扩展为执行更新提供了功能强大的高效机制。然而，在某些情况下，它们会导致非直觉的行为。如果在用户应用程序的设计中未考虑到的这些扩展的使用，则此行为可能会导致异常（例如丢失更新）。

`ansi_update_constraints` 选项控制更新是否限于 SQL/2003 标准允许的范围。

如果将此选项设置为 `Strict`，则无法进行以下更新：

- 对包含 JOINS 的游标的更新
- 对出现在 ORDER BY 子句中的列的更新
- 不允许在 UPDATE 语句中出现 FROM 子句

如果将此选项设置为 `Cursors`，则相同的限制会生效，但只用于游标。如果游标打开时未设置为 `FOR UPDATE` 或 `FOR READ ONLY`，则数据库服务器会根据 SQL/2003 标准选择可更新。如果 `ansi_update_constraints` 选项设置为 `Cursors` 或 `Strict`，则包含 ORDER BY 子句的游标缺省为 `FOR READ ONLY`；否则，它们缺省为 `FOR UPDATE`。

另请参见

- “UPDATE 语句”一节 《SQL Anywhere 服务器 - SQL 参考》

ansinull 选项 [兼容性]

控制对 NULL 值的解释。

允许值

On、Off

缺省值

On

注释

此选项主要是为与 Transact-SQL (Adaptive Server Enterprise) 兼容而设计的。ansinull 选项会影响比较谓语句与 NULL 常量的计算结果，还会影响为针对 NULL 值进行的分组查询所发出的警告。

ansinull 设置为 On 时，WHERE 或 HAVING 子句或者 On 条件中的所有比较谓语句使用 ANSI 三值逻辑。任何使用 = 或 != 与 NULL 进行的比较的结果都为未知。

将 ansinull 设置为 Off 表示 SQL Anywhere 对以下四个条件使用二值逻辑：

expr = NULL

expr != NULL

expr = @var // @var 是过程变量或主机变量

expr != @var

在每种情况下，谓语句都计算为 true 或者 false—从不会为未知。在这样的比较中，NULL 值被视为每个域中的特殊值，两个 NULL 值进行相等性 (=) 比较的结果将是 true。注意，表达式 *expr* 必须相对简单，仅引用列、变量和文字；不允许引用子查询和函数。

ansinull 设置为 On 时，包含至少一个 NULL 值的表达式上的任意集合函数（COUNT(*) 除外）都会生成警告 [集合函数中的空值已消除] (SQLSTATE=01003)。ansinull 设置为 Off 时，不会显示此警告。

限制

- 将 ansinull 设置为 Off 只影响 SELECT、UPDATE、DELETE 和 INSERT 语句中的 WHERE、HAVING 或 ON 谓语句。CASE 或 IF 语句或者 IF 表达式中的比较语义不受影响。
- Adaptive Server Enterprise 12.5 引入了 LIKE 谓语句的行为更改：ansinull 设置为 Off 时使用 NULL 模式字符串。在 SQL Anywhere 中，LIKE 谓语句依旧不受 ansinull 设置的影响。

auditing 选项 [数据库]

在数据库中启用或禁用审计。

允许值

On、Off

缺省值

Off

作用域

只能为 PUBLIC 组设置。设置立即生效。需要 DBA 权限。

注释

此选项开启或关闭审计。

审计是事务日志中有关数据库内多个事件的详细信息的记录。审计以性能的降低为代价，提供了一些安全功能。为数据库开启审计时，不能停止使用事务日志。关闭事务日志之前必须关闭审计。不能以只读模式启动开启了审计的数据库。

为使审计选项起作用，必须将审计选项设置为 On，并使用 `sa_enable_auditing_type` 系统过程指定要审计的信息的类型。审计在以下任一条件为 true 时不会生效：

- auditing 选项设置为 Off
- 审计选项已禁用

如果将 auditing 选项设置为 On 但不指定 auditing 选项，将记录所有类型的审计信息。或者，您可以选择记录以下任意组合：使用 `sa_enable_auditing_type` 系统过程的权限检查、连接尝试、DDL 语句、公共选项和触发器。

另请参见

- “[审计数据库活动](#)”一节第 1012 页
- “[sa_enable_auditing_type 系统过程](#)”一节 《SQL Anywhere 服务器 - SQL 参考》
- “[sa_disable_auditing_type 系统过程](#)”一节 《SQL Anywhere 服务器 - SQL 参考》

示例

启用审计

```
SET OPTION PUBLIC.auditing = 'On';
```

auditing_options 选项 [数据库]

此选项保留供系统使用。请不要更改此选项的设置。

background_priority 选项 [数据库] [不建议使用]

不建议使用。限制对当前连接之外的其它连接的性能的影响。

允许值

On、Off

缺省值

Off

作用域

可以为单个连接或 PUBLIC 组设置此选项。设置立即生效。

如果临时设置此选项，则该设置只适用于当前连接。对于此选项，同一用户 ID 的不同连接可以拥有不同的设置。

background_priority 设置为 on 时，查询内并行机制不用于连接。请参见“[查询执行期间的并行](#)”一节《[SQL Anywhere 服务器 - SQL 的用法](#)》。

注释

将此选项设置为 On 会导致在后台优先级级别执行请求。当将此选项设置为 Off 时，会按照 [优先级] 选项指定的值执行请求。

另请参见

- “[priority 选项 \[数据库\]](#)”一节第 535 页
- “[max_priority 选项 \[数据库\]](#)”一节第 518 页

blob_threshold 选项 [SQL Remote]

控制消息代理作为一个长对象 (BLOB) 来处理的值的长度大小。

允许值

以字节为单位的整数

缺省值

256

注释

任何长度超过 blob_threshold 选项的值都会作为 BLOB 进行复制。即，先将它分成片段，接下来以块的形式复制，然后再在接收者站点使用 SQL 变量重组并将这些片段连接起来。

每条 SQL 语句的大小都必须能放在一个消息中，所以不应将此选项的值设置为大于消息大小（缺省值为 50 KB）。

另请参见

- “[SQL Remote 选项](#)”一节《[SQL Remote](#)》

blocking 选项 [数据库]

控制响应锁定冲突的行为。

允许值

On、Off

缺省值

On

作用域

可以为单个连接或 PUBLIC 组设置此选项。设置立即生效。

注释

将 blocking 选项设置为 On 后，如果事务试图获取锁定，而该锁定与另一事务包含的现有锁定冲突，则该事务要等待所有冲突的锁定都释放或达到 blocking_timeout。如果该锁定在 blocking_timeout 毫秒的时间内没有被释放，则会向正在等待的事务返回错误。如果将 blocking 选项设置为 Off，试图获取冲突锁定的事务会收到错误。

另请参见

- [“blocking_timeout 选项 \[数据库\]”一节第 487 页](#)

blocking_timeout 选项 [数据库]

控制事务获取锁定要等待的时间。

允许值

整数（以毫秒为单位）

缺省值

0

作用域

可以为单个连接或 PUBLIC 组设置此选项。设置立即生效。

注释

当 blocking 选项设置为 On 时，如果事务试图获取与现有锁定冲突的锁定，则该事务将为冲突锁定的释放等待 blocking_timeout 指定的以毫秒为单位的时间。如果该锁定在 blocking_timeout 毫秒的时间内没有被释放，则会向正在等待的事务返回错误。

将此选项设置为 0，会强制所有试图获取锁定的事务等待所有冲突的事务释放其锁定。

另请参见

- [“blocking 选项 \[数据库\]”一节第 486 页](#)

chained 选项 [兼容性]

控制在缺少 BEGIN TRANSACTION 语句时的事务模式。

允许值

On、Off

缺省值

On

对 Open Client 和 jConnect 连接为 Off

注释

控制 Transact-SQL 事务模式。在非链接模式 (chained=Off) 中，除非执行显式的 BEGIN TRANSACTION 语句来启动事务，否则每个语句都是单独提交的。在链接模式 (chained=On) 中，事务是在数据检索或修改语句之前隐式地执行的。

checkpoint_time 选项 [数据库]

设置数据库服务器在不执行检查点操作时可运行的最大分钟数。

允许值

整数

缺省值

60

作用域

只能为 PUBLIC 组设置。需要 DBA 权限。必须关闭并重新启动数据库服务器，才能使更改生效。

注释

此选项与 recovery_time 选项一起使用，以决定应在何时进行检查点操作。

另请参见

- [“了解检查点日志”一节第 17 页](#)
- [“recovery_time 选项 \[数据库\]”一节第 538 页](#)
- [“数据库服务器如何确定何时进行检查点操作”一节第 853 页](#)

cis_option 选项 [数据库]

控制远程数据访问的调试信息是否显示在数据库服务器消息窗口中。

允许值

0, 7

缺省值

0

作用域

可以为单个连接或 PUBLIC 组设置此选项。

注释

此选项控制在使用远程数据访问时，是否在数据库服务器消息窗口中显示有关如何对远程数据库执行查询的信息。将此选项设置为 7 可在数据库服务器消息窗口看到调试信息。当此选项设置为 0（缺省值）时，远程数据访问的调试信息不会显示在数据库服务器消息窗口中。

一旦开启远程跟踪功能，跟踪信息就会出现在数据库服务器消息窗口中。通过在启动数据库服务器时指定 `-o` 服务器选项，您可以将此输出记录到文件中。请参见“[-o 服务器选项](#)”一节第 196 页。

cis_rowset_size 选项 [数据库]

设置每次读取从远程服务器返回的行数。

允许值

整数

缺省值

50

作用域

可以为单个连接或 PUBLIC 组设置此选项。在与远程服务器建立新连接时生效。

注释

此选项会在使用 ODBC 连接远程数据库服务器时设置 ODBC FetchArraySize 的值。

close_on_endtrans 选项 [兼容性]

控制在事务结束时关闭游标。

允许值

On、Off

缺省值

On

jConnect 连接为 Off

注释

当 `close_on_endtrans` 选项设置为 On 时，除非游标在打开时使用了 WITH HOLD，否则游标会在事务提交时被关闭。事务回退时的行为受 `ansi_close_cursors_on_rollback` 选项控制。

在 `close_on_endtrans` 选项设置为 Off 时，无论 `ansi_close_cursors_on_rollback` 选项为何设置，或游标在打开时是否使用 WITH HOLD，游标都不会在提交或回退时关闭。

将此选项设置为 Off，可提供与 Adaptive Server Enterprise 兼容的行为。

另请参见

- [“ansi_close_cursors_on_rollback 选项 \[兼容性\]”](#) 一节第 481 页

collect_statistics_on_dml_updates 选项 [数据库]

在执行修改数据的 DML 语句（例如 INSERT、DELETE 和 UPDATE）期间控制统计信息的收集。

允许值

On、Off

缺省值

On

注释

数据库服务器会在语句正常执行期间更新统计信息，并使用收集到的统计信息自行调整列统计信息。将 `collect_statistics_on_dml_updates` 选项设置为 Off 可在执行修改数据的 DML 语句（例如 INSERT、DELETE 和 UPDATE）期间禁用统计信息的更新。

正常情况下，无需关闭此选项。但是，在频繁更改大量数据的环境中，将此选项设置为 Off 可提高性能—假定 `update_statistics` 选项也设置为 On。

`collect_statistics_on_dml_updates` 选项与 `update_statistics` 选项之间的差别在于：`update_statistics` 选项将满足某个谓语的实际行数与预估满足该谓语的行数进行比较，然后更新估计值。

`collect_statistics_on_dml_updates` 选项根据插入、更新或删除的特定行的值修改列统计信息。

另请参见

- [“update_statistics 选项 \[数据库\]”](#) 一节第 558 页
- [“更新列统计信息以提高优化程序性能”](#) 一节 《SQL Anywhere 服务器 - SQL 的用法》

compression 选项 [SQL Remote]

设置 SQL Remote 消息的压缩级别。

允许值

从 -1 到 9 之间的整数

缺省值

6

注释

这些值具有以下含意：

- **-1** 以版本 5 的格式发送消息。版本 5 的 [消息代理] 不能读取版本 6 和更高版本的 [消息代理] 发送的消息。在您的系统中的所有 [消息代理] 都升级到版本 6 或更高版本之前，应确保将 `compression` 选项设置为 -1。
- **0** 不压缩。
- **1 到 9** 压缩程度递增。创建压缩程度高的消息会比创建压缩程度低的消息花费更多时间。

另请参见

- [“SQL Remote 选项”一节 《SQL Remote》](#)

conn_auditing 选项 [数据库]

控制 `auditing` 选项设置为 On 时为每个连接启用审计还是禁用审计。

允许值

On、Off

缺省值

On

作用域

只能设置为临时选项，保持到当前连接结束。需要 DBA 权限。

注释

只有在登录过程（由 `login_procedure` 数据库选项指定）中设置了 `conn_auditing` 选项时才会生效。将 `conn_auditing` 设置为 On 可为连接启用审计。但是，只有 `auditing` 选项也设置为 On 时才会记录审计信息。可执行以下语句确定是否要审计连接：

```
SELECT CONNECTION_PROPERTY ( 'conn_auditing' );
```

另请参见

- [“控制审计”一节第 1012 页](#)
- [“auditing 选项 \[数据库\]”一节第 484 页](#)
- [“login_procedure 选项 \[数据库\]”一节第 513 页](#)

connection_authentication 选项 [数据库]

指定验证字符串，用于对照经过验证的应用程序的数据库签名验证应用程序签名。

允许值

字符串

缺省值

空字符串

作用域

仅可以为单个连接设置此选项。

注释

只有在使用 SQL Anywhere OEM 版数据库服务器时，此选项才生效。

建立连接后，经过验证的应用程序必须立即为每个连接设置 `connection_authentication` 数据库选项。如果验证签名，则验证连接，除了 SQL 权限强加的限制之外，连接的行为不受任何限制。如果不验证签名，连接的操作仅限于未经验证的应用程序允许的操作。

必须仅使用 `TEMPORARY` 关键字为当前连接的持续时间设置 `connection_authentication` 选项。以下 SQL 语句验证连接：

```
SET TEMPORARY OPTION connection_authentication =  
    'company = company-name;  
    application = application-name;  
    signature = application-signature';
```

`company-name` 和 `application-name` 必须与数据库验证语句中的相应项匹配。`application-signature` 是您从 Sybase 获得的应用程序签名。

如果公司名称中有引号、撇号或其它特殊字符，则这些字符在字符串中必须成对出现才能由系统接受。

有关配置和使用 OEM 版 SQL Anywhere 的详细信息，请参见“[运行经过验证的 SQL Anywhere 应用程序](#)”一节第 70 页。

另请参见

- “[database_authentication \[数据库\]](#)”一节第 495 页

示例

以下示例将指定包含特殊字符的验证字符串：

```
SET TEMPORARY OPTION connection_authentication=  
    'Company = Joe's Garage;  
    Application = Joe's Program;  
    Signature = 0fa55157edb8e14d818e...';
```

continue_after_raiserror 选项 [兼容性]

控制 RAISERROR 语句后面跟随的行为。

允许值

On、Off

缺省值

On

注释

RAISERROR 语句用在过程和触发器中以生成错误。当此选项设置为 Off 时，每当遇到 RAISERROR 语句时，就停止执行过程或触发器。

如果将 `continue_after_raisererror` 选项设置为 On，则 RAISERROR 语句不再发出执行结束的错误。过程会存储 RAISERROR 状态代码和消息，并在结束时返回最近的 RAISERROR。如果导致 RAISERROR 的过程是从其它过程调用的，则在最外层的调用过程结束之前，不会返回 RAISERROR。

在过程结束后，中间 RAISERROR 状态和代码会丢失。如果返回时出现一个错误，则返回新错误的信息，而 RAISERROR 信息丢失。应用程序可以通过在不同的执行点检查 @@error 全局变量来查询中间 RAISERROR 状态。

只有在 `on_tsql_error` 选项设置为 Conditional（缺省设置）时，才使用 `continue_after_raisererror` 选项的设置来控制 RAISERROR 语句后面的行为。如果将 `on_tsql_error` 选项设置为 Stop 或 Continue，`on_tsql_error` 设置会优先于 `continue_after_raisererror` 设置。

另请参见

- “[on_tsql_error 选项 \[兼容性\]](#)” 一节第 528 页

conversion_error 选项 [兼容性]

控制从数据库读取信息时报告数据类型转换错误。

允许值

On、Off

缺省值

On

注释

此选项控制从数据库读取数据或将数据插入数据库时，如果数据类型转换失败，数据库是以错误形式（`conversion_error` 设置为 On）还是以警告形式（`conversion_error` 设置为 Off）进行报告。

当 `conversion_error` 设置为 On 时，生成 `SQL_E_CONVERSION_ERROR` 错误。如果此选项设置为 Off，则生成警告 `SQL_CANT_CONVERT`。

如果只以警告的形式报告转换错误，则使用 NULL 值代替不能进行转换的值。在嵌入式 SQL 中，对于导致错误的列，指示符变量将被设置为 -2。

cooperative_commit_timeout 选项 [数据库]

控制事务日志中 COMMIT 条目写入磁盘的时间。

允许值

整数（以毫秒为单位）

缺省值

250

作用域

可以为单个连接或 PUBLIC 组设置此选项。设置立即生效。

注释

只有在 cooperative_commits 选项设置为 On 时，此选项才有意义。数据库服务器在写入磁盘之前，等待指定的毫秒数，以便其它连接可以填充日志页面。缺省设置为 250 毫秒。

另请参见

- [“cooperative_commits 选项 \[数据库\]”一节第 494 页](#)

cooperative_commits 选项 [数据库]

控制提交写入磁盘的时间。

允许值

On、Off

缺省值

On

作用域

可以为单个连接或 PUBLIC 组设置此选项。设置立即生效。

注释

如果 cooperative_commits 选项设置为 Off，则当数据库服务器接收到 COMMIT 时，会将其写入磁盘，然后允许应用程序继续执行。

如果 cooperative_commits 选项设置为 On（缺省值），且有其它活动连接，则数据库服务器不会立即将 COMMIT 写入磁盘。应用程序会等待一段时间（最长为 cooperative_commit_timeout 选项设置的最大时间值），以便在页面写入磁盘之前其它内容可放置于页面上。

将 cooperative_commits 设置为 On，并且增大 cooperative_commit_timeout 设置，可以削减磁盘 I/O 数，从而增加数据库服务器的总体吞吐量，但代价是各个连接的周转时间会更长。

如果 cooperative_commits 和 delayed_commits 选项都设置为 On，并且 cooperative_commit_timeout 间隔内没有页面写入，则应用程序会继续执行（就如同提交已经生效），并且会使用剩余间隔 (delayed_commit_timeout - cooperative_commit_timeout) 作为 delayed_commits 间隔。然后，会写入页面，即使它们并未填满。

另请参见

- [“delayed_commits 选项 \[数据库\]”](#) 一节第 501 页

database_authentication [数据库]

设置数据库的验证字符串。

允许值

字符串

缺省值

空字符串

作用域

只能为 PUBLIC 组设置。必须重新启动数据库才能使此选项生效。

注释

只有在使用 SQL Anywhere OEM 版数据库服务器时，此选项才生效。

验证数据库后，只有指定了正确验证签名的连接才能在改数据库上执行操作。未验证的连接在只读模式下运行。如果您想使用经过验证的数据库，则必须使用 OEM 版 SQL Anywhere。

要验证数据库，请为数据库设置 `database_authentication` 选项：

```
SET OPTION PUBLIC.database_authentication =  
  'company = company-name;  
  application = application-name;  
  signature = database-signature!';
```

`company-name` 和 `application-name` 参数是您获取签名时提供给 Sybase 的值，而 `database-signature` 是您从 Sybase 收到的数据库签名。

如果公司名称中有引号、撇号或其它特殊字符，则这些字符在字符串中必须成对出现系统才能接受。

数据库服务器装载经过验证的数据库时，它在数据库服务器消息窗口中显示一条消息，说明经过验证的公司和应用程序。可以通过检查此消息是否存在来验证 `database_authentication` 选项是否生效。该消息具有以下格式：

此数据库授权用于:应用程序:application-name 公司:company-name

可以将验证语句存储到 SQL 脚本文件中，以免重复键入长签名。如果将验证语句存储到 `install-dir\scripts\authenticate.sql` 文件中，则只要创建、重建或升级数据库，就会应用它。

有关配置和使用 OEM 版 SQL Anywhere 的详细信息，请参见 [“运行经过验证的 SQL Anywhere 应用程序”](#) 一节第 70 页。

另请参见

- [“connection_authentication 选项 \[数据库\]”](#) 一节第 491 页

示例

```
SET OPTION PUBLIC.database_authentication =
  'company = MyCompany;
  application = MySQLAnywhereApp;
  signature = 0fa55157edb8e14d818e';
```

date_format 选项 [数据库]

为从数据库检索得来的日期设置格式。

有关控制日期格式的解释的详细信息，请参见“[date_order 选项 \[数据库\]](#)”一节第 497 页。

允许值

字符串

缺省

'YYYY-MM-DD'（这种格式符合 ISO 日期格式规范）

作用域

可以为单个连接或 PUBLIC 组设置此选项。设置立即生效。

注释

格式为使用以下符号的字符串：

符号	说明
<i>yy</i>	两位年份
<i>yyyy</i>	四位数表示的年份
<i>mm</i>	两位月份
<i>mmm</i> [<i>m...</i>]	月份的字符简写形式
<i>d</i>	一位数表示的周中的某天（1 代表星期日，7 代表星期六）
<i>dd</i>	以两位数表示的月中的某天
<i>ddd</i> [<i>d...</i>]	周中某天的字符简写形式
<i>jjj</i>	年中的某天，从 1 到 366

每个符号都会替换为按此设置格式的相应日期值。

如果字符数据为多字节，则每个符号的长度反映字符数而不是字节数。例如，'*mmm*' 符号指定月份的长度为三个字符。

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，可让 SQL Anywhere 为所用语言选择适当的大小写。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。

注意

如果以重新排序日期格式的方式更改 date_format 的设置，请确保更改 date_order 选项以使其反映相同的更改，反之亦然。请参见“[date_order 选项 \[数据库\]](#)”一节第 497 页。

另请参见

- “[time_format 选项 \[兼容性\]](#)”一节第 553 页
- “[timestamp_format 选项 \[兼容性\]](#)”一节第 554 页

示例

下表介绍了 date_format 设置与来自下列语句的输出，执行时间为 2008 年 4 月 14 日，星期一。

```
SELECT CAST( CURRENT DATE AS VARCHAR );
```

date_format	CURRENT DATE
yyyy/mm/dd/ddd	2008/04/14/mon
yyyy/Mm/Dd/ddd	2008/4/14/mon
jjj	105
mmm yyyy	apr 2008
Mmm yyyy	Apr 2008
mm-yyyy	04-2008

date_order 选项 [数据库]

控制日期格式的解释。

有关设置从数据库检索得来的日期格式的详细信息，请参见“[date_format 选项 \[数据库\]](#)”一节第 496 页。

允许值

MDY、YMD、DMY

缺省值

YMD（此格式符合 ISO 日期格式规范）

Open Client 和 jConnect 连接的缺省值设置为 MDY

作用域

可以为单个连接或 PUBLIC 组设置此选项。设置立即生效。

注释

数据库选项 `date_order` 用于确定 10/11/12 是 1912 年 10 月 11 日、1910 年 11 月 12 日还是 1912 年 11 月 10 日。

注意

如果以重新排序日期格式的方式更改 `date_order` 的设置，请确保更改 `date_format` 选项和 `timestamp_format` 选项以反映相同的更改，反之亦然。请参见“[date_format 选项 \[数据库\]](#)”一节第 496 页和“[timestamp_format 选项 \[兼容性\]](#)”一节第 554 页。

debug_messages 选项 [数据库]

控制是否执行包含 DEBUG ONLY 子句的 MESSAGE 语句。

允许值

On、Off

缺省值

Off

注释

使用此选项，可以控制存储过程和触发器（其中包含指定了 DEBUG ONLY 子句的 MESSAGE 语句）中调试消息的行为。缺省情况下，此选项设置为 Off，在执行 MESSAGE 语句时不显示调试消息。通过将 `debug_messages` 设置为 On，可以启用所有存储过程和触发器中的调试消息。

注意

将 `debug_messages` 选项设置为 Off 时，DEBUG ONLY 消息并不占用大量资源，因此在生产系统中，通常可以将这些语句保留在存储过程中。不过，在可能会频繁执行这些语句的地方应慎用它们；否则，它们可能会导致性能略微降低。

另请参见

- “MESSAGE 语句”一节 《SQL Anywhere 服务器 - SQL 参考》

dedicated_task 选项 [数据库]

将请求处理任务专用于处理来自单个连接请求。

允许值

On、Off

缺省值

Off

作用域

只能设置为临时选项，保持到当前连接结束。需要 DBA 权限。

注释

dedicated_task 连接选项设置为 On 时，一个请求处理任务将专门处理一个连接请求。通过启用此选项预先建立连接，可以在数据库服务器意外不响应时，收集有关该服务器状态的信息。

default_dbpace 选项 [数据库]

更改创建表时所在的缺省 dbpace。

允许值

字符串。

缺省值

空字符串

作用域

可以为单个连接或 PUBLIC 组设置此选项。设置立即生效。

注释

对于每个数据库，除了系统（主）dbpace 以外，您还可以创建多达十二个 dbpace。若创建表时未指定 dbpace，则使用由此选项设置命名的 dbpace。如果此选项未设置、设置为空字符串或设置为系统，则使用系统 dbpace。

当您创建临时表或索引时，它们始终置于 TEMPORARY dbpace 中，而无论 default_dbpace 选项的设置如何。如果您在创建基表时指定 IN 子句，则会使用 IN 子句所指定的 dbpace，而非由 default_dbpace 选项指定的 dbpace。

如果所有表都在系统 dbpace 之外的位置创建，则系统 dbpace 仅用于检查点日志和系统表。在您由于性能原因希望将检查点日志放在与其余数据库对象位置不同的单独磁盘上时，这很有用。通过

更改所有 CREATE TABLE 语句以指定 dbspace，或通过创建任何表之前更改此选项，可以将检查点日志放在单独的磁盘中。

另请参见

- [“使用附加 dbspace”一节第 23 页](#)
- [“将不同的文件放在不同的设备上”一节 《SQL Anywhere 服务器 - SQL 的用法》](#)
- [“CREATE DBSPACE 语句”一节 《SQL Anywhere 服务器 - SQL 参考》](#)

示例

在以下示例中，将创建一个名为 MyLibrary 的新 dbspace。然后，将缺省 dbspace 设置为此 MyLibrary dbspace，并在此 MyLibrary dbspace 而不是系统 dbspace 中存储 LibraryBooks 表。

```
CREATE DBSPACE MyLibrary
AS 'c:\\dbfiles\\library.db';
SET OPTION default_dspace = 'MyLibrary';
CREATE TABLE LibraryBooks (
    title CHAR(100),
    author CHAR(50),
    isbn CHAR(30),
);
```

default_timestamp_increment 选项 [数据库] [MobiLink 客户端]

指定为保持列中的值唯一而添加到 TIMESTAMP 类型的列中的微秒数。

允许值

1 到 1000000（含）之间的整数

缺省值

1

作用域

可以为单个连接或 PUBLIC 组设置此选项。设置立即生效。

注释

因为在 SQL Anywhere 中 TIMESTAMP 值精确到小数点后第六位，所以缺省情况下添加 1 微秒（0.000001 秒）以区别两个近乎相同的 TIMESTAMP 值。

某些软件（例如 Microsoft Access）会将 TIMESTAMP 值截断至三位小数，致使有效的比较成为一个问题。可以将 truncate_timestamp_values 选项设置为 On 来指定 SQL Anywhere 存储的小数数值位，以此来保持兼容性。

对于 MobiLink 同步，如果要设置此选项，则必须在执行第一次同步之前进行设置。

另请参见

- [“truncate_timestamp_values 选项 \[数据库\] \[MobiLink 客户端\]”一节第 556 页](#)

delayed_commit_timeout 选项 [数据库]

当 `delayed_commits` 选项设置为 `On` 时，指定应用程序执行 `COMMIT` 和 `COMMIT` 实际写入磁盘之间的最大延迟。

允许值

整数（以毫秒为单位）

缺省值

500

作用域

可以为单个连接或 `PUBLIC` 组设置此选项。设置立即生效。

注释

只有在 `delayed_commits` 设置为 `On` 时，此选项才有意义。它控制事务日志中的 `COMMIT` 条目写入磁盘的时间。如果 `delayed_commits` 设置为 `On`，则数据库服务器等待在 `delayed_commit_timeout` 选项中设置的毫秒数，以便其它连接可在当前页面内容写入磁盘之前填充日志页。请参见“[delayed_commits 选项 \[数据库\]](#)”一节第 501 页。

delayed_commits 选项 [数据库]

确定数据库服务器在执行一条 `COMMIT` 后何时将控制返回应用程序。

允许值

`On`、`Off`

缺省

`Off`（这种情况符合 ISO `COMMIT` 行为）

作用域

可以为单个连接或 `PUBLIC` 组设置此选项。设置立即生效。

注释

此选项设置为 `On` 时，数据库服务器会立即响应 `COMMIT` 语句，而不是一直等到 `COMMIT` 的事务日志条目写入磁盘。此选项设置为 `Off` 时，应用程序必须等到 `COMMIT` 写入磁盘。

当此选项设置为 `On` 时，日志会在日志页已满时写入磁盘，或根据 `delayed_commit_timeout` 选项设置写入磁盘，无论哪个条件先满足。即使事务已提交，但如果系统故障发生在数据库服务器响应 `COMMIT` 之后，但在页写入磁盘之前，也有可能发生事务丢失。将 `delayed_commits` 设置为 `On` 并将 `delayed_commit_timeout` 选项设置为高值，会加快响应时间，但在恢复期间稍有丢失已提交事务的风险。

如果 `cooperative_commits` 和 `delayed_commits` 选项都设置为 `On`，并且在 `cooperative_commit_timeout` 这段时间间隔内没有页面写入，则应用程序会继续执行（如同提交已经

生效)，并且会使用剩余的间隔 (`delayed_commit_timeout - cooperative_commit_timeout`) 作为 `delayed_commits` 间隔，在此间隔之后，即使页未满，也会将页写入。

另请参见

- “[cooperative_commit_timeout 选项 \[数据库\]](#)” 一节第 493 页
- “[cooperative_commits 选项 \[数据库\]](#)” 一节第 494 页
- “[delayed_commit_timeout 选项 \[数据库\]](#)” 一节第 501 页

delete_old_logs 选项 [MobiLink 客户端] [SQL Remote] [复制代理]

控制在事务日志的事务被复制或同步后，是否删除事务日志。

允许值

On、Off、Delay、*n* days

缺省值

Off

注释

此选项由 SQL Anywhere MobiLink 客户端、SQL Remote 和 SQL Anywhere 复制代理使用。缺省设置为 Off。当它设置为 On 时，每个旧事务日志都会在其包含的所有更改均已被成功复制或同步时被删除。当它设置为 DELAY 时，如果旧事务日志的文件名指示该日志创建于当前日期，则不会删除这样的事务日志，即使所有更改均已发送并确认。当它设置为 *n* days 时，将删除 *n* 天之前创建的日志。

有关如何将 `delete_old_logs` 选项与 BACKUP 语句一起使用来删除事务日志旧副本的详细信息，请参见“[BACKUP 语句](#)”一节《[SQL Anywhere 服务器 - SQL 参考](#)》。

示例

如果在 1 月 18 日针对 `delete_old_logs` 选项设置为 10 天的远程数据库运行 `dbmlsync`，`dbmlsync` 将删除 1 月 7 日或之前创建的脱机事务日志。远程数据库将如下设置该选项：

```
SET OPTION delete_old_logs = '10 days';
```

另请参见

- “[SQL Anywhere 客户端记录](#)” 一节《[MobiLink - 客户端管理](#)》
- “[MirrorLogDirectory \(mld\) 扩展选项](#)” 一节《[MobiLink - 客户端管理](#)》
- “[SQL Remote 选项](#)” 一节《[SQL Remote](#)》

escape_character 选项 [兼容性]

此选项保留供系统使用。请不要更改此选项的设置。

exclude_operators 选项 [数据库]

此选项保留供系统使用。请不要更改此选项的设置。

extended_join_syntax 选项 [数据库]

控制对于使用多表连接的重复相关名的查询，是允许还是报告为错误。

允许值

On、Off

缺省值

On

注释

如果此选项设置为 On，则 SQL Anywhere 允许在外连接提供空值的一端使用重复的相关名。使用同一相关名指定的所有表或视图被解释为表或视图的同一实例。

以下的 FROM 子句说明了 SQL Anywhere 对使用重复相关名的连接的解释，其中 C1 和 C2 是搜索条件：

```
( R left outer join T on ( C1 ), T join S on ( C2 ) )
```

如果此选项设置为 On，则对此连接的解释如下所示：

```
( R left outer join T on ( C1 ) ) join S on ( C2 )
```

如果此选项设置为 Off，则生成以下错误：

SQL Anywhere 错误 -137:表 'T' 需要唯一的相关名。

注意

查看消除重复相关名的结果，可以使用 REWRITE 函数并将第二个参数设置为 ANSI 来查看重写的语句。

另请参见

- “REWRITE 函数 [Miscellaneous]” 一节 《SQL Anywhere 服务器 - SQL 参考》

external_remote_options [SQL Remote]

指出应将消息链接参数存储在哪里。

允许值

On、Off

缺省值

Off

注释

此选项由 SQL Remote 使用，用来指示消息链接参数应存储在数据库中 (Off)，还是应外部存储 (On)。

fire_triggers 选项 [兼容性]

控制触发器是否在数据库中激发。

允许值

On、Off

缺省值

On

注释

当设置为 On 时，会触发触发器。当设置为 Off 时，则不会触发触发器，参照完整性触发器（例如级联更新和删除）也包括在内。只有具有 DBA 权限的用户才可以设置此选项。此选项会被 -gf 选项替换，后者不管 fire_triggers 设置如何，都会禁止所有触发器的触发。

在从 Adaptive Server Enterprise 向 SQL Anywhere 复制数据时会涉及此选项，因为来自 Adaptive Server Enterprise 事务日志的所有操作（包括由触发器执行的操作）都会复制到 SQL Anywhere。

另请参见

- [“-gf 服务器选项”一节第 180 页](#)
- [“触发器简介”一节《SQL Anywhere 服务器 - SQL 的用法》](#)

first_day_of_week 选项 [数据库]

设置一周中某天的编号。

允许值

1, 2, 3, 4, 5, 6, 7

缺省

7（星期日是一周中的第一天）

注释

允许值有下列含义：

值	含义
1	星期一
2	星期二
3	星期三
4	星期四
5	星期五
6	星期六
7	星期日

此选项指定的值会影响 DATEPART 函数在获取工作日值时的结果。您也可以在 SET 语句中使用 DATEFIRST 选项更改每周的第一天。

此选项指定的值不影响 DOW 函数的结果。例如，即使每周的第一天设置为星期一，对星期一，DOW 函数也返回 2。

另请参见

- “DATEPART 函数 [Date and time]” 一节 《SQL Anywhere 服务器 - SQL 参考》
- “SET 语句 [T-SQL]” 一节 《SQL Anywhere 服务器 - SQL 参考》

for_xml_null_treatment 选项 [数据库]

控制对使用 FOR XML 子句的查询中的 NULL 值的处理。

允许值

Empty、Omit

缺省

Omit

注释

执行包含 FOR XML 子句的查询时，for_xml_null_treatment 选项决定 NULL 值的处理方式。缺省情况下，结果中会忽略包含 NULL 值的元素和属性。将此选项设置为 Empty 会在值为 NULL 时生成空元素或属性。

另请参见

- “使用 FOR XML 子句以 XML 格式检索查询结果” 一节 《SQL Anywhere 服务器 - SQL 的用法》
- “SELECT 语句” 一节 《SQL Anywhere 服务器 - SQL 参考》

force_view_creation 选项 [数据库]

此选项为系统所保留。请不要更改此选项的设置。

小心

只能将 force_view_creation 选项用于 reload.sql 脚本中。此选项由卸载实用程序 (dbunload) 使用，不应对其进行显式设置。

global_database_id 选项 [数据库]

控制使用 DEFAULT GLOBAL AUTOINCREMENT 创建的列的值范围。用于在复制环境下生成唯一主键。

允许值

非负整数

缺省值

2147483647

作用域

只能为 PUBLIC 组设置。需要 DBA 权限。

注释

为此选项指定的值是起始值。对于使用 DEFAULT GLOBAL AUTOINCREMENT 创建的列，当将一个不包括 DEFAULT GLOBAL AUTOINCREMENT 列值的行插入表中时，数据库服务器会为此列生成一个值。该值由 global_database_id 值和列的分区大小决定。

将 global_database_id 设置为缺省值表示禁用 DEFAULT GLOBAL AUTOINCREMENT。在此情况下，NULL 会生成缺省值。

您可以使用以下语句查找当前数据库中选项的值：

```
SELECT DB_PROPERTY( 'GlobalDBID' );
```

此功能在复制环境中用于确保主键唯一时尤为有用。

另请参见

- “CREATE TABLE 语句”一节 《SQL Anywhere 服务器 - SQL 参考》
- GlobalDBID 属性：“数据库属性”一节第 601 页
- MobiLink：“设置全局数据库 ID”一节 《MobiLink - 服务器管理》
- SQL Remote：“重复主键错误”一节 《SQL Remote》
- “重装带有自动增量列的表”一节 《SQL Anywhere 11 - 更改和升级》

示例

以下示例将数据库标识号设置为 100。

```
SET OPTION PUBLIC.global_database_id = '100';
```

http_session_timeout 选项 [数据库]

指定客户端放弃之前等待 HTTP 会话超时的时间（以分钟为单位）。

允许值

整数（1 到 525600）

缺省

30

作用域

只能为 PUBLIC 组设置。需要 DBA 权限。

注释

此选项为 Web 服务应用程序提供可变的会话超时控制。Web 服务应用程序可以在拥有 HTTP 会话的任何请求中更改超时值，但如果 HTTP 会话超时，更改超时值会影响后续等待队列中的请求。Web 应用程序必须包括用来检测客户端是否尝试访问已不存在的 HTTP 会话的逻辑。这可以通过检查 SessionCreateTime 连接属性值以确定时间戳是否有效来完成：如果 HTTP 请求与当前 HTTP 会话不相关联，则 SessionCreateTime 连接属性包含空字符串。

如果需要保留一个连接以供在 HTTP 会话的持续时间内使用，建议在 sa_set_http_option 系统过程中使用 SessionTimeout 选项。请参见“sa_set_http_option 系统过程”一节《SQL Anywhere 服务器 - SQL 参考》。

另请参见

- SessionTimeout、SessionCreateTime 和 http_session_timeout 属性：“连接属性”一节第 566 页
- “sa_set_http_option 系统过程”一节《SQL Anywhere 服务器 - SQL 参考》
- “使用 HTTP 会话”一节《SQL Anywhere 服务器 - 编程》

integrated_server_name 选项 [数据库]

指定域控制器服务器的名称，此服务器用于查找集成登录的 Windows 用户组成员资格。

允许值

字符串

缺省值

NULL

作用域

只能为 PUBLIC 组设置。需要 DBA 权限。

注释

当使用 Windows 用户组进行集成登录时，具有 DBA 权限的用户可以使用此选项来指定用于查找组成员资格的域控制器服务器的名称。缺省情况下，使用运行 SQL Anywhere 的计算机来验证组成员资格。

另请参见

- “为 Windows 用户组创建集成登录” 一节第 102 页
- “GRANT 语句” 一节 《SQL Anywhere 服务器 - SQL 参考》

示例

以下示例指定在计算机 server-1 上验证组成员资格。

```
SET OPTION PUBLIC.integrated_server_name = '\\server-1';
```

isolation_level 选项 [数据库] [兼容性]

控制锁定隔离级别。

允许值

0、1、2、3、快照、语句快照、只读语句快照

缺省值

0

1 (Open Client、jConnect 和 TDS 连接)

作用域

可以为单个连接或 PUBLIC 组设置此选项。设置立即生效。

注释

此选项控制锁定隔离级别，如下所示：

- **0** 允许脏读、非重复性读取和幻像行。
- **1** 阻止进行脏读。允许非重复读取和幻像行。
- **2** 阻止进行脏读和非重复读取。允许幻像行。
- **3** 可串行读。阻止脏读、非重复性读取和幻像行。
- **快照** 从事务读取或更新第一行时开始，使用已提交数据的快照。
- **语句快照** 对于每个语句，从读取数据库的第一行时开始，使用已提交数据的快照。非重复读取和幻像行可能发生在事务中，但不会发生在单个语句中。
- **只读语句快照** 对于只读语句，从读取数据库的第一行时开始，使用已提交数据的快照。非重复读取和幻像行可能发生在事务中，但不会发生在单个语句中。对于可更新语句，使用 `updatable_statement_isolation` 选项指定的隔离级别（可以是 0（缺省值）、1、2 或 3）。

有关支持的隔离级别的更多详细信息，请参见“[隔离级别和一致性](#)”一节《[SQL Anywhere 服务器 - SQL 的用法](#)》。

`allow_snapshot_isolation` 选项必须设置为 `On` 才能使用快照、语句快照或只读语句快照设置。

如果您正在使用 `iAnywhere JDBC` 驱动程序，则缺省隔离级别为 `0`。

在快照、语句快照或只读语句快照隔离级别运行的查询会看到数据库已提交状态的快照。

通过在 `INSERT`、`UPDATE`、`DELETE`、`SELECT`、`UNION`、`EXCEPT` 和 `INTERSECT` 语句中包括 `OPTION` 子句，可以替代此选项在各个语句中的任何临时或公共设置。请参见：

- “[INSERT 语句](#)”一节《[SQL Anywhere 服务器 - SQL 参考](#)》
- “[UPDATE 语句](#)”一节《[SQL Anywhere 服务器 - SQL 参考](#)》
- “[DELETE 语句](#)”一节《[SQL Anywhere 服务器 - SQL 参考](#)》
- “[SELECT 语句](#)”一节《[SQL Anywhere 服务器 - SQL 参考](#)》
- “[UNION 子句](#)”一节《[SQL Anywhere 服务器 - SQL 参考](#)》
- “[EXCEPT 子句](#)”一节《[SQL Anywhere 服务器 - SQL 参考](#)》
- “[INTERSECT 子句](#)”一节《[SQL Anywhere 服务器 - SQL 参考](#)》

另请参见

- “[allow_snapshot_isolation 选项 \[数据库\]](#)”一节第 478 页
- “[updatable_statement_isolation 选项 \[数据库\]](#)”一节第 557 页
- “[快照隔离](#)”一节《[SQL Anywhere 服务器 - SQL 的用法](#)》
- “[隔离级别和一致性](#)”一节《[SQL Anywhere 服务器 - SQL 的用法](#)》
- “[选择隔离级别](#)”一节《[SQL Anywhere 服务器 - SQL 的用法](#)》

java_location 选项 [数据库]

指定数据库的 Java VM 路径。

允许值

字符串

缺省值

空字符串

作用域

只能为 `PUBLIC` 组设置。需要 `DBA` 权限。

注释

缺省情况下，此选项包含一个空字符串。在此情况下，数据库服务器在 `JAVA_HOME` 环境变量、路径和其它位置搜索 Java VM。JavaVM 数据库属性允许您查询在未设置 `java_location` 选项时数据库服务器将使用哪个 Java VM。

另请参见

- [“java_main_userid 选项 \[数据库\]” 一节第 510 页](#)
- [“java_vm_options 选项 \[数据库\]” 一节第 510 页](#)
- JavaVM 属性: [“数据库属性” 一节第 601 页](#)
- [“选择 Java VM” 一节 《SQL Anywhere 服务器 - 编程》](#)

java_main_userid 选项 [数据库]

指定其连接可用于安装类和其它与 Java 相关的管理任务的数据库用户。

允许值

字符串

缺省

DBA 用户（数据库初始化时创建的缺省用户）

作用域

只能为 PUBLIC 组设置。需要 DBA 权限。

注释

指定的用户 ID 应具有 DBA 权限，这样他们才能执行所需的操作。不需要该用户的口令。

另请参见

- [“java_location 选项 \[数据库\]” 一节第 509 页](#)
- [“java_vm_options 选项 \[数据库\]” 一节第 510 页](#)
- [“选择 Java VM” 一节 《SQL Anywhere 服务器 - 编程》](#)

java_vm_options 选项 [数据库]

指定数据库服务器启动 Java VM 时使用的命令行选项。

允许值

字符串

缺省值

空字符串

作用域

只能为 PUBLIC 组设置。需要 DBA 权限。

注释

此选项允许您指定启动 `java_location` 选项指定的 Java VM 时数据库服务器使用的选项。这些附加选项可用于为调试目的而设置 Java VM，或在 Unix 平台上作为服务运行。在某些情况下，以 64 位模式而非 32 位模式使用 Java VM 时也需要附加选项。

另请参见

- [“java_location 选项 \[数据库\]”一节第 509 页](#)
- [“java_main_userid 选项 \[数据库\]”一节第 510 页](#)
- [“选择 Java VM”一节《SQL Anywhere 服务器 - 编程》](#)

示例

以下示例使用 `java_vm_options` 选项在数据库服务器作为服务启动并且用户需要注销时使 Java VM 在 Unix 上保持运行：

```
SET OPTION PUBLIC.java_vm_options = '-Xrs';
```

以下示例指示 Java VM 在 HP-UX 上使用 64 位模式：

```
SET OPTION PUBLIC.java_vm_options = '-d64';
```

log_deadlocks 选项 [数据库]

控制是打开还是关闭死锁报告功能。

允许值

On、Off

缺省值

Off

作用域

只能为 PUBLIC 组设置。需要 DBA 权限。设置立即生效。

注释

当此选项设置为 On 时，数据库服务器在内部缓冲区中记录有关死锁的信息。缓冲区大小是固定的，即 10000 字节。您可以使用 `sa_report_deadlocks` 存储过程来查看死锁信息。当此选项设置为 Off 时，缓冲区的内容会被清除。

发生死锁后，仅报告死锁中涉及的那些连接的信息。报告连接的顺序取决于连接等待行的情况。对于线程死锁，将报告有关所有连接的信息。

死锁报告功能打开时，您还可以使用 Deadlock 系统事件在发生死锁时执行操作。请参见 [“了解系统事件”一节第 868 页](#)。

另请参见

- “sa_report_deadlocks 系统过程”一节 《SQL Anywhere 服务器 - SQL 参考》
- “确定被阻塞的连接”一节 《SQL Anywhere 服务器 - SQL 的用法》
- “教程：诊断死锁”一节 《SQL Anywhere 服务器 - SQL 的用法》

login_mode 选项 [数据库]

控制数据库的集成和 Kerberos 登录的使用。

允许值

一个或多个以下值：Standard、Integrated、Kerberos、Mixed（不建议使用）

缺省

Standard

作用域

只能为 PUBLIC 组设置。需要 DBA 权限。设置立即生效。

注释

此选项指定是否允许标准、集成和 Kerberos 登录。接受以下一种或多种登录模式（这些值区分大小写）：

- **Standard** 允许标准登录。这是缺省设置。标准连接登录必须提供用户 ID 和口令，且不使用集成或 Kerberos 连接参数。
- **Integrated** 允许集成登录。
- **Kerberos** 允许 Kerberos 登录。
- **Mixed（不建议使用）** 这等效于指定 [Standard]、[Integrated]。

如果指定多个登录模式，则数据库服务器允许所有的指定模式。

小心

将 login_mode 数据库选项设置为不允许标准登录可限制只有被授予了集成或 Kerberos 登录映射的用户或组才能进行连接。尝试使用用户 ID 和口令连接会生成错误。只有具有 DBA 权限的用户不受此限制。

可在以逗号分隔的列表中指定多个值。此列表不能包含空格。例如，以下设置既允许标准登录又允许集成登录：

```
SET OPTION PUBLIC.login_mode = 'Standard,Integrated';
```

如果数据库文件未受保护可由未授权的用户复制，则应使用临时公共 login_mode 选项（用于集成登录和 Kerberos 登录）。这样，如果文件被复制，缺省情况下将不再支持集成登录和 Kerberos 登录。

另请参见

- “使用集成登录”一节第 99 页
- “Kerberos 验证”一节第 107 页
- “安全问题：复制的数据库文件”一节第 115 页

示例

仅启用集成登录（标准登录和 Kerberos 登录失败）：

```
SET OPTION PUBLIC.login_mode = 'Integrated';
```

启用标准登录和 Kerberos 登录（集成登录失败）：

```
SET OPTION PUBLIC.login_mode = 'Standard,Kerberos';
```

启用标准、集成和 Kerberos 登录：

```
SET OPTION PUBLIC.login_mode = 'Standard,Integrated,Kerberos';
```

login_procedure 选项 [数据库]

指定一个在启动时设置连接兼容性选项的登录过程。

允许值

字符串

缺省值

sp_login_environment 系统过程

作用域

需要 DBA 权限。

注释

此登录过程在运行时调用 sp_login_environment 过程来判断数据库连接设置。执行完所有检查以验证连接有效后，调用该登录过程。对于事件连接，不执行 login_procedure 选项指定的过程。

通过新建过程并设置 login_procedure 来调用新过程，可自定义缺省数据库选项设置。此自定义过程需要调用 sp_login_environment 或在检测到 TDS 连接时（参见缺省的 sp_login_environment 代码），直接调用 sp_tsql_environment。否则可能会使基于 TDS 的连接不能工作。请不要编辑 sp_login_environment 或 sp_tsql_environment。

可由用户定义的登录过程发出口令已到期错误消息 (SQLSTATE 08WA0) 的信号，向用户指明其口令已到期。发出错误信号可使应用程序检查该错误并处理到期的口令。建议使用登录策略来实现口令到期功能，不建议使用会返回口令已过期错误消息的登录过程。

如果使用 NewPassword=* 连接参数，则需要对客户库发出此错误信号，提示输入新口令。如果过程发出 SQLSTATE 28000（无效用户 ID 或口令）或 SQLSTATE 08WA0（已过期的口令）信号，或者过程通过 RAISERROR 报告错误，则登录失败并向用户返回一条错误消息。如果发出任何其它错误信号或发生另外错误，则用户登录成功并向数据库服务器消息日志写入一条消息。

另请参见

- “[post_login_procedure 选项 \[数据库\]](#)” 一节第 531 页
- “[sp_login_environment 系统过程](#)” 一节 《[SQL Anywhere 服务器 - SQL 参考](#)》
- “[sp_tsqf_environment 系统过程](#)” 一节 《[SQL Anywhere 服务器 - SQL 参考](#)》
- “[提高口令安全性](#)” 一节第 1006 页
- “[NewPassword 连接参数 \[NEWPWD\]](#)” 一节第 274 页
- “[CREATE PROCEDURE 语句 \(Web 服务\)](#)” 一节 《[SQL Anywhere 服务器 - SQL 参考](#)》

示例

以下示例显示了如何通过发出 `INVALID_LOGON` 错误信号禁止一个连接。

```
CREATE PROCEDURE DBA.login_check( )
  BEGIN
    DECLARE INVALID_LOGON EXCEPTION FOR SQLSTATE '28000';
    // Allow a maximum of 3 concurrent connections
    IF( DB_PROPERTY( 'ConnCount' ) > 3 ) THEN
      SIGNAL INVALID_LOGON;
    ELSE
      CALL sp_login_environment;
    END IF;
  END
go

GRANT EXECUTE ON DBA.login_check TO PUBLIC
go

SET OPTION PUBLIC.login_procedure='DBA.login_check'
go
```

有关禁止连接的其它方法的详细信息，请参见 “[RAISERROR 语句](#)” 一节 《[SQL Anywhere 服务器 - SQL 参考](#)》。

以下示例显示了用户在 30 分钟的时间段内连接失败次数超过 3 次时，您如何阻止连接尝试。受阻周期内的所有受阻尝试都会接收到无效的口令错误，并被记录为失败。该日志会保留足够长的时间以供 DBA 对它进行分析。

```
CREATE TABLE DBA.ConnectionFailure(
  pk INT PRIMARY KEY DEFAULT AUTOINCREMENT,
  user_name CHAR(128) NOT NULL,
  tm TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP
)
go

CREATE INDEX ConnFailTime ON DBA.ConnectionFailure(
  user_name, tm )
go

CREATE EVENT ConnFail TYPE ConnectFailed
HANDLER
BEGIN
  DECLARE usr CHAR(128);
  SET usr = event_parameter( 'User' );

  // Put a limit on the number of failures logged.
  IF (SELECT COUNT(*) FROM DBA.ConnectionFailure
    WHERE user_name = usr
    AND tm >= DATEADD( minute, -30,
      CURRENT_TIMESTAMP )) < 20 THEN
```

```

        INSERT INTO DBA.ConnectionFailure( user_name )
            VALUES( usr );
        COMMIT;
        // Delete failures older than 7 days.
        DELETE DBA.ConnectionFailure
        WHERE user_name = usr
        AND tm < dateadd( day, -7, CURRENT_TIMESTAMP );
        COMMIT;
    END IF;
END
go

CREATE PROCEDURE DBA.login_check( )
BEGIN
    DECLARE usr CHAR(128);
    DECLARE INVALID_LOGON EXCEPTION FOR SQLSTATE '28000';
    SET usr = CONNECTION_PROPERTY( 'userid' );
    // Block connection attempts from this user
    // if 3 or more failed connection attempts have occurred
    // within the past 30 minutes.
    IF ( SELECT COUNT( * ) FROM DBA.ConnectionFailure
        WHERE user_name = usr
        AND tm >= DATEADD( minute, -30,
            CURRENT_TIMESTAMP ) ) >= 3 THEN
        SIGNAL INVALID_LOGON;
    ELSE
        CALL sp_login_environment;
    END IF;
END
go

GRANT EXECUTE ON DBA.login_check TO PUBLIC
go

SET OPTION PUBLIC.login_procedure='DBA.login_check'
go

```

下面的示例显示如何发出 [口令已到期] 消息。建议使用登录策略来实现口令到期通知。

```

CREATE PROCEDURE DBA.check_expired_login( )
BEGIN
    DECLARE PASSWORD_EXPIRED EXCEPTION FOR SQLSTATE '08WA0';

    IF( condition-to-check-for-expired-password ) THEN
        SIGNAL PASSWORD_EXPIRED;
    ELSE
        CALL sp_login_environment;
    END IF;
END;

```

有关登录策略的信息，请参见“[管理登录策略概述](#)”一节第 416 页。

materialized_view_optimization 选项 [数据库]

控制优化程序如何使用实例化视图来有效回答查询。

允许值

Disabled、**Fresh**、**Stale**、 $N \{ \text{Minute[s]} | \text{Hour[s]} | \text{Day[s]} | \text{Week[s]} | \text{Month[s]} \}$

缺省值

Stale

作用域

可以为单个连接、单个用户或 PUBLIC 组设置此选项。设置立即生效。

注释

materialized_view_optimization 选项允许您指定优化程序可以在哪些情况下使用失效的实例化视图。

如果实例化视图所引用的任何基表中的数据得到更新，则实例化视图中的数据将失效。确定实例化视图的刷新频率时，应考虑数据失效的可接受程度以及刷新视图需要花费的时间，因为刷新过程中该视图不能用于查询。还应考虑数据库服务器返回的结果可能不反映数据库当前状态时，该结果是否可接受。可为此选项选择以下设置：

- **Disabled** 不将实例化视图用于查询优化。
- **Fresh** 仅当实例化视图为最新时（自上次刷新视图以来未修改基础表中的数据）才使用它。
- **Stale** 即使实例化视图已失效也使用它们。这是缺省设置。
- **N { Minute[s] | Hour[s] | Day[s] | Week[s] | Month[s] }** 只要在指定的时间段内已刷新失效的实例化视图，就使用最新和失效的实例化视图。指定的分钟值必须小于 2^{31} 分钟。数据库服务器将一周按 7 天处理，将一个月按 30 天处理。

当查询直接引用实例化视图时，不论是否失效都使用该视图；此时 materialized_view_optimization 选项不起作用。

max_client_statements_cached 选项 [数据库]

控制由客户端高速缓存的语句数。

允许值

整数，0 到 100

作用域

可以为单个连接或 PUBLIC 组设置此选项。更改值操作立即生效。

缺省值

10

说明

多次准备相同的 SQL 语句时，客户端语句高速缓存可减少数据库请求数和语句准备数。重复准备和删除相同的 SQL 文本时，客户端可高速缓存该语句，使其在数据库服务器上保留为已准备语句，即使它已由应用程序删除。高速缓存语句可节省数据库服务器删除和重新准备语句的额外工作量。如果模式更改发生、数据库选项设置更改或执行 DROP VARIABLE 语句，则自动删除已准备语句，并在下次执行 SQL 语句时再次准备，确保决不重新使用可能造成错误行为的高速缓存语句。

此选项指定可保留为已准备（高速缓存）的语句的最大数。对于 `max_statement_count` 资源控制器，高速缓存语句数不计算在内。

此选项的设置适用于使用嵌入式 SQL、ODBC、OLE DB、ADO.NET 和 iAnywhere JDBC 驱动程序进行的连接。它不适用于 Open Client、jConnect 或 HTTP 连接。

将此选项设置为 0 会禁用客户端语句高速缓存。如果应用程序重复准备和删除超过 10 个相同 SQL 语句，则增加此值可能提高性能。例如，如果应用程序循环遍历二十五个 SQL 语句，在循环的每个迭代都要准备和删除这些 SQL 语句，其中的每个语句在每个迭代都具有完全相同的文本，则将此选项设置为 25 可以提高性能。

但增加此选项的值会增加客户端上占用的内存，也会给数据库服务器施加更大高速缓存压力。如果大量高速缓存语句由于模式更改或选项设置不能重新使用，则会自动禁用连接的语句高速缓存。如果自动关闭语句高速缓存，则客户端会定期再次打开语句高速缓存，重新评估决定，确定重新启用语句高速缓存是否有益。

另请参见

- [“max_statement_count 选项 \[数据库\]” 一节第 520 页](#)
- ClientStmtCacheHits 和 ClientStmtCacheMisses 属性：[“连接属性” 一节第 566 页](#)
- ClientStmtCacheHits 和 ClientStmtCacheMisses 属性：[“数据库服务器属性” 一节第 588 页](#)

max_cursor_count 选项 [数据库]

对限制连接一次能使用的最大游标数的资源调控器进行控制。

允许值

整数

缺省值

50

作用域

可以为单个连接或 PUBLIC 组设置此选项。设置立即生效。为*任何*连接设置此选项，都需要有 DBA 权限。

注释

此资源调控器允许 DBA 限制用户可使用的每个连接的游标数。如果某个操作超出连接限制，则会生成一个错误，指示超出资源的调控器范围。

如果连接执行一个存储过程，则该过程的执行受过程所有者权限的控制。然而，由该过程所用的资源却被指派给当前连接。

您可以通过将此选项设置为 0（零）来取消资源限制。

max_plans_cached 选项 [数据库]

指定将存储在高速缓存中的执行计划的最大数目。

允许值

整数

缺省值

20

作用域

可以为单个连接或 PUBLIC 组设置此选项。设置立即生效。必须拥有 DBA 权限才能为 PUBLIC 组设置此选项。

注释

此选项指定为每个连接而高速缓存的计划的数目。对于在存储过程、函数和触发器内执行的查询、INSERT、UPDATE 和 DELETE 语句，优化程序会为它们缓存执行计划。在存储过程、存储函数或触发器中的语句由某个连接执行几次之后，优化程序会为该语句建立可重用的计划。

可重用的计划不会将主机变量值用于选择性估计或重写优化。因此，使用可重用的计划可能会比重新优化语句的开销要高一些。当可重用的计划的开销接近于语句的最佳观测开销时，优化程序会将该计划添加到计划高速缓存中。

在执行修改表模式的语句（例如 CREATE TABLE 和 DROP TABLE 语句）后，高速缓存会被清除。引用已声明的临时表的语句不会被高速缓存。

将此选项设置为 0 会禁用计划高速缓存。

另请参见

- “计划高速缓存”一节 《SQL Anywhere 服务器 - SQL 的用法》
- max_plans_cached 属性：“连接属性”一节第 566 页

max_priority 选项 [数据库]

控制连接的最大优先级级别。

允许值

Critical、High、Above Normal、Normal、Below normal、Low、Background

缺省值

Normal

作用域

可以为单个连接或 PUBLIC 组设置此选项。设置立即生效。需要 DBA 权限。

如果临时设置此选项，则该设置只适用于当前连接。对于此选项，同一用户 ID 的不同连接可以拥有不同的设置。

注释

无论请求的优先级级别如何，不同优先级级别的调度都允许所有请求获得一些 CPU 时间。优先级较高的请求获得的时间片比低级别的多。

另请参见

- [“priority 选项 \[数据库\]”一节第 535 页](#)

max_query_tasks 选项 [数据库]

指定数据库服务器在以并行操作方式处理一个查询时可使用的最大服务器任务数。

允许值

整数

缺省值

0

作用域

可以为单个连接或 PUBLIC 组设置此选项。设置立即生效。

注释

max_query_tasks 选项设置任何 SQL 语句都可以使用的最大并行操作级别。该选项设置以并行操作方式处理一个查询时可使用的数据库服务器任务数。缺省值为 0，允许数据库服务器使用它选择的任意多的任务。max_query_tasks 选项的任何其它值都设置每个查询允许的最大任务数。将 max_query_tasks 选项设置为 1 会禁用查询内并行机制。

有关服务器任务、线程和查询执行的详细信息，请参见 [“SQL Anywhere 中的线程”一节第 47 页](#)和 [“设置数据库服务器的进程并发水平”一节第 50 页](#)。

数据库服务器可用于所有请求的任务数受启动时使用 -gn 选项设置的阈值限制。对于由该服务器提供服务的所有数据库和连接，此数值是全局最大值。用于一个请求的任务数也受数据库服务器可用的逻辑处理器数目的限制。例如，使用 -gtc 选项将处理器并发数设置为 1 会禁用查询内并行机制。

启用时，查询内并行机制用于处理符合某些限定条件的 SELECT 语句。在查询的访问计划中如果存在交换运算符，则表明使用了查询内并行机制。

通过在 INSERT、UPDATE、DELETE、SELECT、UNION、EXCEPT 和 INTERSECT 语句中包括 OPTION 子句，可以替代此选项在各个语句中的任何临时或公共设置。请参见：

- [“INSERT 语句”一节 《SQL Anywhere 服务器 - SQL 参考》](#)
- [“UPDATE 语句”一节 《SQL Anywhere 服务器 - SQL 参考》](#)
- [“DELETE 语句”一节 《SQL Anywhere 服务器 - SQL 参考》](#)
- [“SELECT 语句”一节 《SQL Anywhere 服务器 - SQL 参考》](#)
- [“UNION 子句”一节 《SQL Anywhere 服务器 - SQL 参考》](#)
- [“EXCEPT 子句”一节 《SQL Anywhere 服务器 - SQL 参考》](#)
- [“INTERSECT 子句”一节 《SQL Anywhere 服务器 - SQL 参考》](#)

另请参见

- “-gn 服务器选项” 一节第 182 页
- “-gt 服务器选项” 一节第 185 页
- “-gtc 服务器选项” 一节第 186 页
- “查询执行期间的并行” 一节 《SQL Anywhere 服务器 - SQL 的用法》
- max_query_tasks 属性: “数据库服务器属性” 一节第 588 页

max_recursive_iterations 选项 [数据库]

限制递归公用表表达式可用的最大迭代次数。

允许值

整数

缺省值

100

作用域

可以为单个连接或 PUBLIC 组设置此选项。设置立即生效。必须拥有 DBA 权限才能为 PUBLIC 组设置此选项。

注释

如果递归公用表表达式的计算未能在指定的迭代次数内完成，则计算会中止，并产生一条错误消息。递归子查询在每增加一次迭代时所需的资源量通常会呈几何增长。设置此选项后，可限制在检测到无穷递归之前所消耗的时间和资源量，但仍允许递归公用表表达式按预期方式工作。

将此选项设置为 0 将禁用递归公用表表达式。

另请参见

- “公用表表达式” 《SQL Anywhere 服务器 - SQL 的用法》

max_statement_count 选项 [数据库]

对限制某个连接能同时使用的经过准备的语句数的资源调控器进行控制。

允许值

整数

缺省值

50

作用域

可以为单个连接或 PUBLIC 组设置此选项。设置立即生效。为*任何*连接设置此选项，都需要有 DBA 权限。

注释

如果不再需要预准备的语句时它们未能显式删除，则使用预准备语句的应用程序可能接收到错误 [超出 '经过准备的语句数' 的资源调控器]。`max_statement_count` 数据库选项是一个资源调控器，它允许 DBA 限制每个连接使用的预准备的语句数。如果某个操作超出连接限制，则会生成一个错误，指示超出资源的调控器范围。

如果连接执行一个存储过程，则该过程的执行受过程所有者权限的控制。然而，由该过程所用的资源却被指派给当前连接。

数据库服务器为连接创建的每个预准备语句维护所需的数据结构。只有在应用程序通知数据库服务器不再需要预准备语句或连接断开时才释放这些结构。要减少连接的语句计数，必须执行与 DROP STATEMENT 请求等效的命令。下表列出可针对 SQL Anywhere 支持的 API 执行的命令：

接口	语句
ADO	RecordSet.Close
ADO.NET	SADaReader.Close 或 SADaReader.Dispose
嵌入式 SQL	DROP STATEMENT
Java	resultSet.Close、Statement.Close
ODBC	SQLFreeStmt(hstmt, SQL_DROP) 或 SQLFreeHandle(SQL_HANDLE_STMT, hstmt)

注意

在 Java 和 .NET 中，建议您显式删除语句。不应依赖垃圾收集来执行此清除任务，因为语言例程不发出服务器调用以释放语句资源。此外，也不能保证垃圾收集例程何时执行。

如果服务器在任一时间对任一连接需要支持的预准备语句数超过缺省值，则应将 `max_statement_count` 设置为更高的值。但是，请注意，大量的活动预准备语句会消耗更多服务器内存。可以通过将 `max_statement_count` 选项设置为 0（零）来完全禁用预准备语句资源调控器，但建议不这样做。对于未适当释放预准备语句的应用程序，这样做会使数据库服务器易于因内存不足而关闭。

另请参见

- “准备语句”一节 《SQL Anywhere 服务器 - 编程》
- “DROP STATEMENT 语句 [ESQL]”一节 《SQL Anywhere 服务器 - SQL 参考》

max_temp_space 选项 [数据库]

控制一个连接可以使用的最大临时文件空间。

允许值

Integer [**k** | **m** | **g** | **p**]

缺省值

0

作用域

可以设置为当前连接的持续时间或 PUBLIC 组的临时选项。设置立即生效。需要 DBA 权限。

注释

此选项允许您指定在请求因超出临时文件空间限制而失败之前一个连接可以使用的最大临时文件空间。要使 `max_temp_space` 选项生效，必须将 `temp_space_limit_check` 选项设置为 On（缺省）。

缺省值 0 表示一个连接可以请求的临时文件空间没有固定限制。任何其它值均指定一个连接可以使用的临时文件空间的字节数。可以分别使用 **k**、**m** 或 **g** 将单位指定为千字节、兆字节或千兆字节。如果使用 **p**，该参数为可用临时文件总空间的百分比。

对于请求临时文件空间的连接，数据库服务器针对 `max_temp_space` 选项的设置检查限制，以确保请求的大小在最大值以下。如果连接请求多于允许量的临时空间，则请求失败并生成错误 `SQLSTATE_TEMP_SPACE_LIMIT`。

可以为单个连接或 PUBLIC 组设置此选项。设置立即生效。必须拥有 DBA 权限才能为 PUBLIC 组设置此选项。

另请参见

- [“temp_space_limit_check 选项 \[数据库\]” 一节第 552 页](#)
- [“sa_disk_free_space 系统过程” 一节 《SQL Anywhere 服务器 - SQL 参考》](#)

示例

为连接设置 1 GB 限制:

```
SET OPTION PUBLIC.max_temp_space = '1g';
```

以下两个语句都为连接设置 1 MB 限制:

```
SET OPTION PUBLIC.max_temp_space = 1048576;
```

```
SET OPTION PUBLIC.max_temp_space = '1m';
```

使用总可用临时空间的 5%:

```
SET OPTION PUBLIC.max_temp_space = '5p';
```

min_password_length 选项 [数据库]

为数据库中的新口令设置最小长度。

允许值

整数

值以字节为单位。对于单字节字符集，它与字符数相同。

缺省值

0 个字符

作用域

只能为 PUBLIC 组设置。设置立即生效。需要 DBA 权限。

注释

此选项允许数据库管理员为实现更高的安全性而对所有新口令强制使用一个最小长度。现有口令不受影响。口令的最大长度为 255 字节并且区分大小写。

另请参见

- [“verify_password_function 选项 \[数据库\]”一节第 560 页](#)

示例

将新口令的最小长度设置为 6 个字节。

```
SET OPTION PUBLIC.min_password_length = 6;
```

nearest_century 选项 [兼容性]

控制字符串到日期的转换中对两位的年份的解释。

允许值

0 到 100（含）之间的整数

缺省值

50

注释

此选项控制在从字符串转换为日期或时间戳时对两位数年份的处理。

nearest_century 设置为一个用作切换点的数字值。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。

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，这样对于那些 yy 值小于 50 的年份就会被设置为 20yy。

non_keywords 选项 [兼容性]

关闭各个关键字，从而允许它们用作标识符。

允许值

字符串

缺省值

空字符串

注释

此选项将关闭单个关键字。这提供了一种确保使用本软件的旧版本创建的应用程序不会因新关键字而无法工作的方法。如果在数据库中有一个标识符，而它现在是一个关键字，则可以在所有应用程序或脚本中将该标识符用双引号引上，或者使用 `non_keywords` 选项关闭此关键字。

以下语句防止将 `TRUNCATE` 和 `SYNCHRONIZE` 识别为关键字：

```
SET OPTION non_keywords = 'TRUNCATE, SYNCHRONIZE';
```

此选项的每个新设置都会替换以前的设置。以下语句会清除所有以前的设置。

```
SET OPTION non_keywords =;
```

此选项的一个副作用是不能使用具有已关闭关键字的 SQL 语句：它们会生成语法错误。

另请参见

- “关键字”一节 《SQL Anywhere 服务器 - SQL 参考》

odbc_describe_binary_as_varbinary [数据库]

控制 SQL Anywhere ODBC 驱动程序描述 BINARY 列的方式。

允许值

On、Off

缺省值

Off

注释

此选项允许您选择是将所有 BINARY 和 VARBINARY 列作为 BINARY 还是作为 VARBINARY 提供给应用程序。缺省情况下，SQL Anywhere ODBC 驱动程序将 BINARY 和 VARBINARY 列均描述为 `SQL_BINARY`。当此选项设置为 On 时，ODBC 驱动程序将 BINARY 和 VARBINARY 列描述为 `SQL_VARBINARY`。无论此选项如何设置，均无法区分 BINARY 和 VARBINARY 列。

如果使用 Delphi 应用程序，则 BINARY 列始终用零填充，而 VARBINARY 列不用零填充，此时将此选项设置为 On 可能会很有用。通过将此选项设置为 On，使所有列均被视为可变长度数据类型，可以提高 Delphi 的性能。

另请参见

- “BINARY 数据类型”一节 《SQL Anywhere 服务器 - SQL 参考》
- “VARBINARY 数据类型”一节 《SQL Anywhere 服务器 - SQL 参考》

odbc_distinguish_char_and_varchar 选项 [数据库]

控制 SQL Anywhere ODBC 驱动程序描述 CHAR 列的方式。

允许值

On、Off

缺省值

Off

注释

当某个连接打开时，SQL Anywhere ODBC 驱动程序使用此选项的设置来确定描述 CHAR 列的方式。如果此选项设置为 Off（缺省值），则 CHAR 列描述为 SQL_VARCHAR。如果此选项设置为 On，则 CHAR 列描述为 SQL_CHAR。VARCHAR 列始终都描述为 SQL_VARCHAR。

odbc_distinguish_char_and_varchar 选项还控制 NCHAR 列是描述为 SQL_WCHAR 还是 SQL_WVARCHAR。如果此选项设置为 Off，则 NCHAR 列描述为 SQL_WVARCHAR。如果此选项设置为 On，则 NCHAR 列描述为 SQL_WCHAR。NVARCHAR 列始终都描述为 SQL_WVARCHAR。

另请参见

- “NCHAR 数据类型”一节 《SQL Anywhere 服务器 - SQL 参考》
- “NVARCHAR 数据类型”一节 《SQL Anywhere 服务器 - SQL 参考》

oem_string 选项 [数据库]

在数据库文件头页中存储用户指定的信息。

允许值

字符串（最多 128 字节）

缺省值

空字符串

作用域

只能为 PUBLIC 组设置。设置立即生效。需要 DBA 权限。

注释

可以在数据库文件头页中存储信息，以便以后通过从您的应用程序直接读取文件来抽取该信息。此页面存储在系统 dbospace 文件头中。如果为 OEM 字符串指定的值超过 128 字节，则会返回一个错误。

您可能会发现存储这类信息（例如模式版本、应用程序名称、应用程序版本等）很有用。或者，不启动数据库，应用程序可以使用 OEM 字符串确定数据库文件是否与应用程序相关联，或将应用程

序设计为使用此信息确认数据库文件可供应用程序使用，方法是在使用数据库文件之前存储应用程序为进行验证而读取的字符串。您还可以抽取元数据以显示给用户。

要在系统 `dbspace` 文件头中设置 `oem_string`，请执行以下语句：

```
SET OPTION PUBLIC.oem_string=user-specified-string;
```

user-specified-string 值存储在 `ISYSOPTIONS` 系统表和系统 `dbspace` 文件标题中。在 `SET OPTION` 语句中指定字符串之前，必须在所需的字符集中定义该字符串，因为在 `SET OPTION` 语句中提供该字符串时，对该字符串不进行任何转换。可以使用 `CSCONVERT` 函数将字符串转换为所需的字符集。

您可以通过以下方法查询 `oem_string` 的值：

- 使用 `oem_string` 连接属性：

```
SELECT CONNECTION_PROPERTY( 'oem_string' );
```

- 使用 `SYSOPTION` 系统视图：

```
SELECT setting FROM SYSOPTION WHERE "option" = 'oem_string';
```

◆ 从应用程序查询 `oem_string` 选项

1. 打开数据库系统 `dbspace` 文件。
2. 将文件的第一页读取到缓冲区。
3. 在缓冲区中搜索分别在 `OEM` 字符串前后的两字节前缀和后缀序列。

前缀和后缀字符串在 `sqldef.h` 中分别定义为 `DB_OEM_STRING_PREFIX` 和 `DB_OEM_STRING_SUFFIX`。这两个字符串之间的所有字节就是在数据库中定义的 `OEM` 字符串。

SQL Anywhere 在 `oem_string` 目录中包括两个示例程序：

- `dboem.cpp` 为 C 程序，它说明如何抽取 `OEM` 字符串并将其打印到数据库服务器消息窗口。
- `dboem.pl` 说明如何抽取 `OEM` 字符串并将其打印到 PERL 脚本内的 `stdout`。

小心

应用程序不能在数据库中直接写入 `OEM` 字符串，因为这会损坏数据库文件头页。

在 Windows 中，服务器已加载数据库文件时，应用程序不能直接读取该文件。数据库服务器在该文件上有一个独占锁。但是，在任何受支持的 Unix 平台上，具有读权限的应用程序可以在任何时候直接读取该文件。对 `OEM` 字符串的更改不会在该文件中立即显示出来。发出检查点操作会使数据库服务器将页面 0 刷新到磁盘，并反映当前 `OEM` 字符串值。

如果在更改 `OEM` 字符串和下一个检查点操作之间数据库服务器失败，则文件头不会反映新 `OEM` 字符串值；新 `OEM` 字符串值将在数据库经过恢复后正确设置。

另请参见

- “[CSCONVERT 函数 \[String\]](#)” 一节 《SQL Anywhere 服务器 - SQL 参考》

示例

以下示例将包含有关数据库文件信息的 OEM 字符串进行加密并将其存储在数据库头文件中：

```
BEGIN
  DECLARE @v VARCHAR(100);
  SET @v = BASE64_ENCODE( ENCRYPT( 'database version 10', 'abc' ) );
  EXECUTE IMMEDIATE 'SET OPTION PUBLIC.oem_string = '' || @v || ''';
END;
```

您可以使用以下命令检索 OEM 字符串的值：

```
SELECT DECRYPT(
  BASE64_DECODE(
    CONNECTION_PROPERTY( 'oem_string' ) ), 'abc' )
```

on_charset_conversion_failure 选项 [数据库]

控制在字符转换期间遇到错误时要采取的操作。

允许值

Ignore、Warning、Error

缺省

Ignore

注释

控制在字符转换期间遇到错误时要采取的操作，如下所示：

- **Ignore** 不会出现错误和警告。
- **Warning** 将替代和非法字符报告为警告。不转换非法字符。
- **Error** 将替代和非法字符报告为错误。

客户端和数据库之间需要字符集转换时，此选项控制当检测到非法字符或使用字符替换时是忽略、返回警告还是返回错误。

单字节到单字节的转换器无法报告替代和非法字符，且必须设置为 Ignore。

此选项不控制客户端上发生有损耗的转换时的行为。例如，客户端的 SQL 语句必须为或转换为 CHAR 数据库字符集。假设 Unicode 客户端应用程序准备一个 SQL 语句，并且此语句包含不能用 CHAR 数据库字符集表示的字符。将使用替换字符来代替。但由于有损耗的转换发生在客户端上，因此数据库服务器不知道发生了有损耗的转换。

另请参见

- “CHAR 和 NCHAR 之间的比较”一节 《SQL Anywhere 服务器 - SQL 参考》
- “将 NCHAR 转换为 CHAR”一节 《SQL Anywhere 服务器 - SQL 参考》
- “替换字符”一节 《SQL Anywhere 服务器 - SQL 参考》

on_tsq_error 选项 [兼容性]

控制存储过程中的错误处理。

允许值

字符串（有关允许值，请参见以下内容）

缺省值

Conditional

对 JConnect 连接，继续执行

注释

此选项控制存储过程中的错误处理。

- **Stop** 发现错误时立即停止执行。
- **Conditional** 如果过程使用 ON EXCEPTION RESUME，则跟在错误后面的语句处理该错误，继续执行，否则就退出。
- **Continue** 继续执行，无论后面是什么语句。如果有多个错误，则返回在存储过程中所遇到的第一个错误。

使用 on_tsq_error 的 Conditional 和 Continue 设置的目的是为了与 Adaptive Server Enterprise 兼容，而 Continue 可最近似地模拟 Adaptive Server Enterprise 的行为。为了早一些报告错误，请在创建新的 Transact-SQL 存储过程时使用 Conditional 设置。

当此选项设置为 Stop 或 Continue 时，它会取代 continue_after_raiserror 选项的设置。然而，当此选项设置为 Conditional（缺省值）时，RAISERROR 语句后的行为由 continue_after_raiserror 选项的设置决定。

另请参见

- “CREATE PROCEDURE 语句（Web 服务）”一节 《SQL Anywhere 服务器 - SQL 参考》
- “CREATE PROCEDURE 语句 [T-SQL]”一节 《SQL Anywhere 服务器 - SQL 参考》
- “Transact-SQL 过程语言概述”一节 《SQL Anywhere 服务器 - SQL 的用法》
- “continue_after_raiserror 选项 [兼容性]”一节第 492 页

optimization_goal 选项 [数据库]

确定优化查询处理的意图：是迅速返回第一行，还是为最大程度地降低返回整个结果集的开销。

允许值

First-row、All-rows

缺省值

All-rows

注释

optimization_goal 选项控制 SQL Anywhere 针对响应时间还是总资源消耗而优化 SQL 数据操作语言 (DML) 语句。

如果此选项设置为 All-rows (缺省值), 则 SQL Anywhere 将优化查询以选择预计总检索时间最少的访问计划。将 optimization_goal 设置为 All-rows 可能适用于要处理整个结果集的应用程序, 例如 PowerBuilder DataWindow 应用程序。All-rows 设置也适用于不敏感 (ODBC 静态) 游标, 因为当游标打开时整个结果都得以实现。它可能还适用于滚动 (ODBC 键集决定) 游标, 因为这种游标的意图是允许在结果集中滚动。

如果将选项设置为 First-row, SQL Anywhere 会选择用于减少查询结果的首行提取时间访问计划, 而且很可能以总检索时间为代价。特别是, SQL Anywhere 优化程序通常会尽量避免需要实现结果以减少返回首行的时间的访问计划。使用此设置, 优化程序会倾向于利用索引来满足查询的 ORDER BY 子句的访问计划, 而不是需要显式排序操作的计划。

您可以在查询的 FROM 子句中使用 FASTFIRSTROW 表提示, 将特定查询的优化目标设置为 First-row, 而不必更改 optimization_goal 设置。

有关使用 FASTFIRSTROW 表提示的详细信息, 请参见“FROM 子句”一节《SQL Anywhere 服务器 - SQL 参考》。

通过在 INSERT、UPDATE、DELETE、SELECT、UNION、EXCEPT 和 INTERSECT 语句中包括 OPTION 子句, 可以替代此选项在各个语句中的任何临时或公共设置。请参见:

- “INSERT 语句”一节《SQL Anywhere 服务器 - SQL 参考》
- “UPDATE 语句”一节《SQL Anywhere 服务器 - SQL 参考》
- “DELETE 语句”一节《SQL Anywhere 服务器 - SQL 参考》
- “SELECT 语句”一节《SQL Anywhere 服务器 - SQL 参考》
- “UNION 子句”一节《SQL Anywhere 服务器 - SQL 参考》
- “EXCEPT 子句”一节《SQL Anywhere 服务器 - SQL 参考》
- “INTERSECT 子句”一节《SQL Anywhere 服务器 - SQL 参考》

optimization_level 选项 [数据库]

控制 SQL Anywhere 查询优化程序为查找 SQL 语句的访问计划而消耗的资源量。

允许值

0-15

缺省值

9

注释

optimization_level 选项控制 SQL Anywhere 优化程序为优化 SQL 数据操作语言 (DML) 语句而消耗的资源量。此选项控制优化程序将对任何 SELECT 块最多考虑多少种备用连接策略。optimization_level 设置越高, 优化程序考虑的最大连接策略数就会越多。

如果此选项设置为 0，则 SQL Anywhere 优化程序选择它考虑执行的第一个访问计划，从而有效避免对备用计划进行基于开销的比较。此外，如果级别为 0，则禁用某些嵌套查询的语义优化。如果将此选项设置为大于 0 的值，则优化程序会计算备用策略，并选择预计开销最低的策略。如果此选项的设置值大于缺省值 (9)，优化程序将更积极地搜索备用策略，因而可能在优化阶段花费更多的时间。

通常情况下，如果应用程序希望更快地打开 DML 语句，可将此选项临时设置为较低的级别 (0、1 或 2)。我们已经了解，尽管语句可能很复杂，但查询执行时间非常短，因而优化程序选择哪个访问计划关系不大。我们建议您不要更改 optimization_level 的缺省 PUBLIC 设置。

设置 optimization_level 选项的效果与 optimization_goal 和 optimization_workload 选项设置无关。

简单 DML 语句（在 WHERE 子句中包含相等条件且唯一地标识特定行的单块、单表查询）均经过启发式优化，并完全跳过基于开销的优化程序。简单 DML 语句的优化不受 optimization_level 选项设置的影响。通过优化程序跳过机制优化的请求数可通过 QueryBypassed 连接属性检索。

有关 QueryBypassed 连接属性的详细信息，请参见“[连接属性](#)”一节第 566 页。

通过在 INSERT、UPDATE、DELETE、SELECT、UNION、EXCEPT 和 INTERSECT 语句中包括 OPTION 子句，可以替代此选项在各个语句中的任何临时或公共设置。请参见：

- “INSERT 语句”一节 《SQL Anywhere 服务器 - SQL 参考》
- “UPDATE 语句”一节 《SQL Anywhere 服务器 - SQL 参考》
- “DELETE 语句”一节 《SQL Anywhere 服务器 - SQL 参考》
- “SELECT 语句”一节 《SQL Anywhere 服务器 - SQL 参考》
- “UNION 子句”一节 《SQL Anywhere 服务器 - SQL 参考》
- “EXCEPT 子句”一节 《SQL Anywhere 服务器 - SQL 参考》
- “INTERSECT 子句”一节 《SQL Anywhere 服务器 - SQL 参考》

optimization_workload 选项 [数据库]

确定优化查询处理的目标是针对更新和读取混合进行的负载还是针对主要基于读取的负载。

允许值

Mixed、OLAP

缺省值

Mixed

作用域

只能为 PUBLIC 组设置。需要 DBA 权限。

注释

optimization_workload 选项控制 SQL Anywhere 是针对更新和读取混合工作负载还是仅针对以读取为主的工作负载对查询进行优化。

如果该选项设置为 Mixed（缺省值），则 SQL Anywhere 选择的查询优化算法将适用于混合有简单插入、更新和删除以及长时间运行的只读查询的工作负载。

如果该选项设置为 OLAP，SQL Anywhere 选择的算法将适用于大部分由长时间运行的查询结合一些批处理更新所组成的工作负载。特别是，优化程序可能选择使用 [聚簇散列分组依据] 查询执行算法。

当该选项设置为 OLAP 时，将启用 [聚簇散列分组依据] 算法。如果该选项设置为 Mixed（缺省值），则禁用该算法。

通过在 INSERT、UPDATE、DELETE、SELECT、UNION、EXCEPT 和 INTERSECT 语句中包括 OPTION 子句，可以替代此选项在各个语句中的任何临时或公共设置。请参见：

- “INSERT 语句”一节 《SQL Anywhere 服务器 - SQL 参考》
- “UPDATE 语句”一节 《SQL Anywhere 服务器 - SQL 参考》
- “DELETE 语句”一节 《SQL Anywhere 服务器 - SQL 参考》
- “SELECT 语句”一节 《SQL Anywhere 服务器 - SQL 参考》
- “UNION 子句”一节 《SQL Anywhere 服务器 - SQL 参考》
- “EXCEPT 子句”一节 《SQL Anywhere 服务器 - SQL 参考》
- “INTERSECT 子句”一节 《SQL Anywhere 服务器 - SQL 参考》

另请参见

- “ClusteredHashGroupBy 算法 (GrByHClust)”一节 《SQL Anywhere 服务器 - SQL 的用法》

pinned_cursor_percent_of_cache 选项 [数据库]

指定多少高速缓存可用于驻留游标。

允许值

0-100 之间的整数

缺省值

10

作用域

只能为 PUBLIC 组设置。需要 DBA 权限。

注释

数据库服务器将多页虚拟内存用于执行游标所需的数据结构。在读取请求之间，这些页会一直锁定在内存中，这样，在下一个读取请求到达时，可迅速获得这些页。

为防止这些页在低内存环境下占用太多高速缓存，可对驻留游标时能使用的高速缓存的百分比设置限制。可以使用 pinned_cursor_percent_of_cache 选项调整该限制。

该选项值指定为一个 0 到 100 之间的百分比，缺省值为 10。将此选项设置为 0 意味着在读取请求之间不会驻留游标页。

post_login_procedure 选项 [数据库]

指定一个结果集中包含消息的过程，这些消息应在用户连接时由应用程序显示。

允许值

字符串

缺省值

post_login_procedure 系统过程

作用域

需要 DBA 权限。

注释

当 post_login_procedure 选项设置为非空字符串的任何值时，应用程序可以调用该选项指定的过程作为连接过程的一部分，以确定应对用户显示什么消息（如果有）。该选项值应为 *owner.function-name* 的形式，以防止用户替换函数。

如果设置了此选项，则 Sybase Central 的 SQL Anywhere 插件、Interactive SQL 和 dbisqlc 会调用该过程，并在窗口中显示该过程返回的任何消息。应修改未随 SQL Anywhere 一起提供的其它应用程序，以调用此选项指定的过程并显示消息（如果需要此功能）。

应用程序可能需要显示有关连接的消息的一种情况是，通知用户其口令即将到期（如果实施了口令到期机制）。使用该功能可在用户的口令实际到期之前，在每次用户连接时通知用户，几天后其口令是否会到期。

此选项指定的过程必须返回包含一行或多行和两列的结果集。类型为 VARCHAR(255) 的第一列返回消息文本，如果没有消息，则返回 NULL。类型为 INT 的第二列返回操作类型。允许的操作值如下：

- **0** 显示消息（如果有）。
- **1** 显示消息并提示用户更改口令。
- **2-99** 保留。
- **100 及更大值** 用户定义。

无论操作值为何，SQL Anywhere 插件、dbisql 和 dbisqlc 均显示所有非 NULL 消息。如果操作设置为 1，则 SQL Anywhere 插件和 dbisql（但不是 dbisqlc）提示用户更改口令，然后将新口令设置为用户指定的值。

有关使用 post_login_procedure 并且包括高级口令规则及实现口令到期功能的示例，请参见 "使用口令验证函数"。

另请参见

- [“login_procedure 选项 \[数据库\]”一节第 513 页](#)
- [“提高口令安全性”一节第 1006 页](#)

示例

以下示例使用名为 `p_post_login_check` 的过程，此过程警告用户其口令即将到期，然后提示用户更改口令。

```
CREATE PROCEDURE DBA.p_post_login_check( )
RESULT( message_text VARCHAR(255), message_action INT )
BEGIN
  DECLARE message_text          CHAR(255);
  DECLARE message_action        INT;

  -- assume the password_about_to_expire variable was
  -- set by the login procedure
  IF password_about_to_expire = 1 THEN
    SET message_text = 'Your password is about to expire';
    SET message_action = 1;
  ELSE
    SET message_text = NULL;
    SET message_action = 0;
  END IF;
  -- return message (if any) through this result set
  SELECT message_text, message_action;
END;

GRANT EXECUTE ON DBA.p_post_login_check TO PUBLIC;

SET OPTION PUBLIC.post_login_procedure = 'DBA.p_post_login_check';
```

precision 选项 [数据库]

指定任意数字算术的结果中数字的最大位数。

允许值

1 到 127（含）之间的整数

缺省值

30

作用域

只能为 PUBLIC 组设置。设置立即生效。

注释

精度是小数点左右两侧的数字总数。scale 选项指定按最大精度值截断算术结果时，小数点后最少有几位数。

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 个数字。如果小数位数为 4，则结果将为 DECIMAL(15.4)。如果小数位数为 2，则结果将为 DECIMAL(15.2)。在这两个示例中都有溢出的可能。

prefetch 选项 [数据库]

控制在客户端应用程序可以使用行之前是否先将行读取到客户端。

允许值

Off、Conditional、Always

缺省值

Conditional

作用域

可以为单个连接或 PUBLIC 组设置此选项。设置立即生效。

注释

此选项控制在客户端应用程序可以使用行之前，是否先将行读取到客户端。即使在客户端应用程序一次请求一行时（例如，当在游标的行上循环时），也一次读取多行，这样做可通过降低对数据库的请求数量而降低响应时间，并提高总吞吐量。

- Off 表示不执行预读。
- Conditional（缺省值）表示执行预读，除非游标类型为 SENSITIVE 或查询包含代理表。
- Always 表示执行预读，即使游标类型为 SENSITIVE 以及游标涉及代理表。

使用 Always 值必须小心，因为它会影响某些游标语义。例如，它会使敏感游标变得不敏感。如果在预取和应用程序读取请求之间更新了值，则可能读取较旧的值。此外，对涉及代理表的游标使用预读可能会产生错误 -668。如果客户端试图重新读取预读行，则游标被限定于 FETCH NEXT 操作。如果在第一次读取列后首次绑定读取列或重新绑定读取列，或在使用 GET DATA 的某些情况下，客户端可能会试图在回退后或在读取相对位置 0 时重新读取预读行。

值敏感游标类型包括 ESQL SENSITIVE 和 SCROLL 游标类型，及 ODBC 和 OLE DB DYNAMIC 与 KEYSSET 游标类型。

prefetch 选项的设置会被 Open Client 和 jConnect 连接忽略。

如果将 DisableMultiRowFetch 连接参数设置为 YES，则会忽略 prefetch 数据库选项，并且不执行任何预读。

此选项以前接受的值为 On。现在此值是 Conditional 的别名。

另请参见

- “预取行”一节 《SQL Anywhere 服务器 - 编程》
- “DisableMultiRowFetch 连接参数 [DMRF]”一节第 263 页

preserve_source_format 选项 [数据库]

控制是否在系统文件中保存过程、触发器、视图和事件处理程序的初始源代码定义。如果保存，则会保存在 SYSTAB、SYSPROCEDURE、SYSTRIGGER 和 SYSEVENT 中的 source 列内。

允许值

On、Off

缺省值

On

作用域

只能为 PUBLIC 组设置。需要 DBA 权限。

注释

当 `preserve_source_format` 为 On 时，数据库服务器会保存来自过程、视图、触发器和事件上的 CREATE 和 ALTER 语句的格式化源代码，并将其置于相应系统视图的 `source` 列。

未格式化的源文本存储在同一个系统表中的 `proc_defn`、`trigger_defn` 和 `view_defn` 列内。然而，这些定义在 Sybase Central 中并不容易阅读。格式化的 `source` 列允许您使用自选的间距、注释和大小写来查看这些定义。

可以将此选项关闭以减少用于保存数据库中对象定义的空间。只能为用户 PUBLIC 设置此选项。

prevent_article_pkey_update 选项 [数据库] [MobiLink 客户端]

控制在发布中涉及的表的主键列的更新。

允许值

On、Off

缺省值

On

注释

将该选项设置为 On 时，禁止对发布中包含的表的主键列进行更新。该选项有助于确保数据完整性，特别是在复制和同步环境中。

小心

强烈建议您不要在同步或复制环境中将此选项设置为 Off。

priority 选项 [数据库]

设置执行来自连接的请求的优先级级别。

允许值

Critical、High、Above Normal、Normal、Below Normal、Low、Background

缺省值

Normal

作用域

可以为单个连接或 PUBLIC 组设置此选项。设置立即生效。

如果临时设置此选项，则该设置只适用于当前连接。对于此选项，同一用户 ID 的不同连接可以拥有不同的设置。

注释

不能将此选项的值设置为高于 max_priority 选项的值。

另请参见

- [“max_priority 选项 \[数据库\]”](#) 一节第 518 页

qualify_owners 选项 [SQL Remote]

控制由 SQL Remote 复制的 SQL 语句是否应使用限定的对象名。

允许值

On、Off

缺省值

On

注释

当在 SQL Anywhere 安装中不需要限定时，消息将比选项设置为 Off 时稍小。

另请参见

- [“SQL Remote 选项”](#) 一节 《SQL Remote》

query_mem_timeout 选项 [数据库]

设置请求等待内存授予的最大时间（以毫秒为单位）。

允许值

-1、0、正整数

缺省值

-1

注释

如果此选项设置为 -1（缺省值）或任何小于 0 的值，则请求等待内存授予的时间最长可达到该请求估计执行时间的 50 倍。如果此选项设置为 0，则请求将始终等待授予内存。如果设置为其它值，该值为请求等待内存授予的最大时间（以毫秒为单位）。

另请参见

- “内存调控器”一节 《SQL Anywhere 服务器 - SQL 的用法》

quote_all_identifiers 选项 [SQL Remote]

控制 SQL Remote 所复制的 SQL 语句是否应使用带引号的标识符。

允许值

On、Off

缺省值

Off

注释

当此选项为 Off 时，dbremote 将 SQL Anywhere 要求使用引号的标识符加上引号（一贯如此）。如果将该选项设置为 On，则所有标识符都加上引号。

另请参见

- “SQL Remote 选项”一节 《SQL Remote》

quoted_identifier 选项 [兼容性]

控制对用双引号引起的字符串的解释。

允许值

On、Off

缺省值

On

对 Open Client 和 jConnect 连接为 Off

注释

此选项控制将双引号内的字符串解释为标识符 (On) 还是解释为文字字符串 (Off)。quoted_identifier 选项是为与 Transact-SQL 兼容而加入的。

请参见 “为实现 Transact-SQL 兼容性设置选项”一节 《SQL Anywhere 服务器 - SQL 的用法》。

read_past_deleted 选项 [数据库]

控制隔离级别 1 和 2 上对未提交删除的服务器行为。

允许值

On、Off

缺省值

On

注释

如果 read_past_deleted 为 On（缺省值），则隔离级别为 1 和 2 的顺序扫描将跳过未提交的已删除行。如果为 Off，则顺序扫描将在隔离级别 1 和 2 上在未提交的已删除行被阻塞（除非删除事务提交或回退）。此选项更改隔离级别为 1 和 2 的服务器行为。

大多数情况下，此选项应保留为 On。如果设置为 Off，阻塞行为将取决于优化程序所选择的计划（如果存在可以使用的索引）。

recovery_time 选项 [数据库]

设置数据库服务器从系统失败中恢复所花费的最长时间，以分钟为单位。

允许值

整数（以分钟为单位）

缺省值

2

作用域

只能为 PUBLIC 组设置。需要 DBA 权限。服务器重新启动后生效。

注释

此选项与 checkpoint_time 选项一起使用，以决定应在何时进行检查点操作。

SQL Anywhere 根据上次检查点操作之后执行的操作，使用启发式算法来估计恢复时间。此恢复时间既包括数据库的估计恢复时间，也包括数据库的估计检查点时间。因此，恢复时间并不精确。

另请参见

- “自动恢复过程”一节第 833 页
- “checkpoint_time 选项 [数据库]”一节第 488 页
- “-gr 服务器选项”一节第 184 页
- “数据库服务器如何确定何时进行检查点操作”一节第 853 页

remote_idle_timeout 选项 [数据库]

控制允许 Web 服务客户端过程和函数处于非活动状态的最大秒数。

允许值

整数（以秒为单位）

缺省值

15

注释

此选项影响 Web 服务客户端过程和函数。如果处于非活动状态的时间超出指定秒数，则过程或函数超时。

replicate_all 选项 [复制代理]

允许整个数据库充当复制服务器安装中的主站点。

允许值

On、Off

缺省值

Off

注释

此选项仅由 SQL Anywhere 复制代理使用。当它设置为 On 时，整个数据库都将设置为充当复制服务器安装中的主站点。对数据库所做的所有更改均由复制代理发送到复制服务器。

另请参见

- [“复制整个数据库”一节第 1091 页](#)

replication_error 选项 [SQL Remote]

允许您指定在出现 SQL 错误时由消息代理调用的存储过程。

允许值

存储过程名

缺省值

无过程

注释

对于 SQL Remote, `replication_error` 选项允许您指定在出现 SQL 错误时由消息代理调用的存储过程。缺省情况下, 不调用任何存储过程。

该过程必须有类型为 CHAR、VARCHAR 或 LONG VARCHAR 的单个参数。SQL 错误消息会调用一次该过程, 并且导致该错误的 SQL 语句也会调用一次该过程。在某些环境中 (例如外键违规), 引发错误的 SQL 语句不可用, 因此只能调用一次该存储过程。

尽管此选项允许您跟踪并监视复制中的 SQL 错误, 但是您仍然必须在您的安装之外设计它们; 此选项并非旨在解决这样的错误。

另请参见

- “SQL Remote 选项” 一节 《SQL Remote》
- “`replication_error_piece` 选项 [SQL Remote]” 一节第 540 页

`replication_error_piece` 选项 [SQL Remote]

与 `replication_error` 选项结合使用允许您指定在 SQL Remote 复制期间出现 SQL 错误时由消息代理调用的 LONG VARCHAR 存储过程。

允许值

存储过程名

缺省值

无过程

注释

如果发生错误并定义了 `replication_error`, 则 `replication_error` 过程以完整的错误字符串来调用。

如果 `replication_error` 和 `replication_error_piece` 均已定义, 则错误会拆分为 VARCHAR 片段。`replication_error` 由第一个片段调用, `replication_error_piece` 由其余片段重复调用。

另请参见

- “`replication_error` 选项 [SQL Remote]” 一节第 539 页
- “SQL Remote 选项” 一节 《SQL Remote》

`request_timeout` 选项 [数据库]

控制单个请求可以运行的最长时间。此选项可用于防止连接长时间消耗大量服务器资源。

允许值

整数, 0 到 86400 (一天), 以秒为单位

缺省值

0

注释

当此选项设置为 0 时，请求不超时。

如果请求花费的时间大约超出 `request_timeout` 秒（挂钟时间而非 CPU 时间），则该请求会中断，并会为用户返回一个错误。返回的错误为 `SQL_E_REQUEST_TIMEOUT: [请求因超时而中断]`。如果请求被阻塞，并且 `blocking_timeout` 选项设置为 0，则在返回阻塞错误（例如 `SQL_E_LOCKED: [用户 '%1' 锁定了 '%2' 中的行]`）之前，该请求最多可保持阻塞 `request_timeout` 秒。

不允许使用用户和公共值 1 到 14。如果连接花费的时间很长（例如，由于复杂的登录过程），则这样做会防止将用户锁在数据库服务器之外。

此选项由数据库客户端和 HTTP/HTTPS 请求两者使用。请注意，在存储过程或 HTTP/HTTPS 请求中设置该选项对当前请求没有影响，因为已使用请求开始时的选项值。

设置 `request_timeout` 公共选项时应小心，因为这可能会使具有长时间运行的请求的应用程序（例如 `dbvalid`、`dbbackup` 和 `dbunload`）失败。此外，设置 `request_timeout` 时，不使用大量服务器资源但能被其他用户阻塞的应用程序也可能失败。解决此类问题的一个方法是，仅对某些应用程序根据连接的 `APPINFO` 值在登录过程中设置相应 `request_timeout` 选项。

如果每个请求计算速度都很快（例如在读取包含很多行的结果集时），则设置此选项可能不会防止应用程序使用大量服务器资源。

另请参见

- [“blocking_timeout 选项 \[数据库\]”一节第 487 页](#)
- [“AppInfo 连接参数 \[APP\]”一节第 249 页](#)

return_date_time_as_string 选项 [数据库]

控制在查询时如何将日期、时间或时间戳值传递给客户端应用程序。

允许值

On、Off

缺省值

Off

作用域

只能设置为临时选项，保持到当前连接结束。

注释

此选项指示日期、时间和时间戳值是作为日期或时间数据类型还是作为字符串返回应用程序。

当此选项设置为 On 时，数据库服务器会在将日期、时间或时间戳值发送到客户端之前将它们转换为字符串，以保持 `timestamp_format`、`date_format` 或 `time_format` 选项设置。

Sybase Central 和 Interactive SQL 会自动将 `return_date_time_as_string` 选项设置为 On。

另请参见

- “`date_format` 选项 [数据库]” 一节第 496 页
- “`time_format` 选项 [兼容性]” 一节第 553 页
- “`timestamp_format` 选项 [兼容性]” 一节第 554 页

rollback_on_deadlock [数据库]

控制当发生死锁时如何处理事务。

允许值

On、Off

缺省值

On

作用域

可以由任何用户设置，并可以针对 PUBLIC 组和单个连接来设置。设置立即生效。

注释

当此选项设置为 On 时，事务遇到死锁时将自动回退。回退在当前请求完成之后进行。如果此选项设置为 Off，SQL Anywhere 将自动回退遇到死锁的语句，并向该事务返回一个错误，指出发生了何种形式的死锁。请注意，回退语句可能并不释放该语句获取的任何锁。

有关死锁的详细信息，请参见“死锁”一节《SQL Anywhere 服务器 - SQL 的用法》。

row_counts 选项 [数据库]

指定数据库是否始终对打开的查询中的行计数。

允许值

On、Off

缺省值

Off

作用域

可以为单个连接或 PUBLIC 组设置此选项。设置立即生效。

注释

如果将此选项设置为 Off，则行计数通常只是一个估计值。如果将此选项设置为 On，则行计数始终是准确值。

小心

当 `row_counts` 设置为 `On` 时，执行查询可能会花费相当长的时间。事实上，它通常会导致 SQL Anywhere 执行两次查询，从而使执行时间加倍。

save_remote_passwords 选项 [SQL Remote]

保存在消息链接中输入的口令。

允许值

On、Off

缺省值

On

注释

如果您要在外部存储消息链接参数，而不是将它们存储在数据库中，则最好不要保存这些口令。您可以通过将此选项设置为 `Off`，避免保存口令。

另请参见

- “SQL Remote 选项”一节 《SQL Remote》

scale 选项 [数据库]

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。

允许值

0 到 127（含）之间的整数，小于为 `precision` 数据库选项指定的值

缺省值

6

作用域

只能为 `PUBLIC` 组设置。设置立即生效。

注释

乘、除、加、减和集合函数都可以有超过最大精度的结果。

另请参见

- “`precision` 选项 [数据库]”一节第 533 页

secure_feature_key [数据库]

允许您使用数据库服务器 -sf 选项启用受保护的连接功能。

允许值

字符串

缺省值

NULL

作用域

只能设置为临时选项，保持到当前连接结束。

注释

通过在启动数据库服务器时包括 -sf 选项，可以指定在服务器上运行的数据库不能使用的功能。-sk 服务器选项允许您指定可用于重新启用连接的所有受保护（已禁用）功能的密钥，并授予该连接更改在数据库服务器上运行的所有数据库的受保护功能的权限。如果将 secure_feature_key 临时选项的值设置为在数据库服务器启动时 -sk 指定的值，则可以为该数据库连接重新启用所有功能，并且在该连接上可以使用 sa_server_option 系统过程控制对数据库功能的访问。

将 secure_feature_key 选项的值设置为任何不同于 -sk 所指定的值时，将不会给出任何错误，由 -sf 指定的功能对该连接仍然禁用。

另请参见

- “-sk 服务器选项” 一节第 210 页
- “-sf 服务器选项” 一节第 207 页
- “sa_server_option 系统过程” 一节 《SQL Anywhere 服务器 - SQL 参考》
- “指定受保护的功能” 一节第 1010 页

示例

以下命令启动一个名为 secure_server 的数据库服务器，启动时将禁用对请求日志的访问权限，并禁用所有远程数据访问功能。可随后使用由 -sk 选项指定的密钥为特定数据库连接启用这些功能。

```
dbsrv11 -n secure_server -sf request_log,remote -sk j978kls12 testdb.db
```

对于在 secure_server 数据库服务器上运行的数据库，如果将 secure_feature_key 选项设置为由 -sk 指定的值，会为该连接启用对请求日志和远程数据访问功能的访问权限：

```
SET TEMPORARY OPTION secure_feature_key = 'j978kls12';
```

sort_collation 选项 [数据库]

允许在 ORDER BY 表达式上隐式使用 SORTKEY 函数。

允许值

Internal、collation_name 或 collation_id

缺省值

Internal

注释

当此选项的值为 **Internal** 时，ORDER BY 子句保持不变。

如果将此选项的值设置为有效的归类名称或归类 ID，则在处理 ORDER BY 子句中的 CHAR 或 NCHAR 字符串表达式时，会像已经调用了 SORTKEY 函数那样处理。不修改使用其它字符串数据类型（如 BINARY、UUID、XML 或 VARBIT）的字符串表达式。

另请参见

- “[SORTKEY 函数 \[String\]](#)” 一节 《[SQL Anywhere 服务器 - SQL 参考](#)》

示例

将排序归类设置为二进制：

```
SET TEMPORARY OPTION sort_collation='binary';
```

将排序归类设置为二进制会转换以下查询：

```
SELECT Name, ID
FROM Products
ORDER BY Name, ID;
SELECT name, ID
FROM Products
ORDER BY 1, 2;
```

查询将被转换为：

```
SELECT Name, ID
FROM Products
ORDER BY SORTKEY(Name, 'binary'), ID;
```

sql_flagger_error_level 选项 [兼容性]

控制对任何未包含在指定标准中的 SQL 进行的响应。

允许值

- Off
- SQL:1992/Entry
- SQL:1992/Intermediate
- SQL:1992/Full
- SQL:1999/Core
- SQL:1999/Package
- SQL:2003/Core
- SQL:2003/Package
- UltraLite

缺省值

Off

注释

此选项将任何未包含在指定标准中的 SQL 标记为错误。例如，指定 SQL:2003/Package 时将导致数据库服务器标记不是完整 SQL/2003 语法的语法。

缺省行为 Off 关闭错误标记。

为了与以前 SQL Anywhere 版本兼容，也接受以下值，并按下面的指定进行映射：

- **E** 此选项对应于 SQL:1992/Entry。
- **I** 此选项对应于 SQL:1992/Intermediate。
- **F** 此选项对应于 SQL:1992/Full。
- **W** 此选项对应于 Off。

另请参见

- “sa_ansi_standard_packages 系统过程”一节 《SQL Anywhere 服务器 - SQL 参考》
- “SQLFLAGGER 函数 [Miscellaneous]”一节 《SQL Anywhere 服务器 - SQL 参考》
- “sql_flagger_warning_level 选项 [兼容性]”一节第 546 页
- “SQL 预处理器”一节 《SQL Anywhere 服务器 - 编程》

sql_flagger_warning_level 选项 [兼容性]

控制对任何未包含在指定标准中的 SQL 进行的响应。

允许值

- Off
- SQL:1992/Entry
- SQL:1992/Intermediate
- SQL:1992/Full
- SQL:1999/Core
- SQL:1999/Package
- SQL:2003/Core
- SQL:2003/Package
- UltraLite

缺省值

Off

注释

此选项将任何未包含在指定标准中的 SQL 标记为警告。例如，指定 SQL:2003/Package 时将导致数据库服务器标记不是完整 SQL/2003 语法的语法。

缺省行为 Off 关闭警告标记。

为了与以前版本兼容，也接受以下值，并按下面的指定进行映射：

- **E** 此选项对应于 SQL:1992/Entry。

- **I** 此选项对应于 SQL:1992/Intermediate。
- **F** 此选项对应于 SQL:1992/Full。
- **W** 此选项对应于 Off。

另请参见

- “sa_ansi_standard_packages 系统过程”一节 《SQL Anywhere 服务器 - SQL 参考》
- “SQLFLAGGER 函数 [Miscellaneous]”一节 《SQL Anywhere 服务器 - SQL 参考》
- “sql_flagger_error_level 选项 [兼容性]”一节第 545 页
- “SQL 预处理器”一节 《SQL Anywhere 服务器 - 编程》

sr_date_format 选项 [SQL Remote]

为从数据库检索得来的日期设置格式。

允许值

由下面所列的符号组成的字符串

缺省值

YYYY/MM/DD

注释

消息代理在复制存储日期的列时使用此选项。格式为使用以下符号的字符串：

符号	说明
YY	两位年份
YYYY	四位数表示的年份
MM	两位月份
MMM[m...]	月份的字符简写形式—有多少个 "m", 就有多少个字符
DD	以两位数表示的月中的某天

每个符号都会替换为按此设置格式的相应日期值。

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，可让 SQL Anywhere 为所用语言选择适当的大小写。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。

如果字符数据为多字节，则每个符号的长度反映字符数而不是字节数。例如，'mmm' 符号指定月份的长度为三个字符。

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：

该选项是根据以下符号生成的字符串：

- 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。

另请参见

- [“sr_time_format 选项 \[SQL Remote\]”一节第 548 页](#)
- [“sr_timestamp_format \[SQL Remote\]”一节第 549 页](#)
- [“SQL Remote 选项”一节《SQL Remote》](#)

sr_time_format 选项 [SQL Remote]

为从数据库中检索得到的时间设置格式。

允许值

由下面所列的符号组成的字符串

缺省值

HH:NN:SS.SSSSS

注释

消息代理在复制存储时间的列时使用此选项。格式为使用以下符号的字符串：

符号	说明
<i>HH</i>	两位小时（24 小时时钟）。
<i>NN</i>	两位数表示的分钟数。
<i>MM</i>	两位分钟（如果跟着冒号，如 hh:mm）。
<i>SS.ssssss</i>	秒及秒的小数部分，共六个小数位。并非所有平台都支持精度达到六位的时间戳。

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也显示为大写。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。

注释

在格式字符串中使用混合大小写将禁止使用前导零。

另请参见

- “[sr_date_format 选项 \[SQL Remote\]](#)” 一节第 547 页
- “[sr_timestamp_format \[SQL Remote\]](#)” 一节第 549 页
- “[SQL Remote 选项](#)” 一节 《SQL Remote》

sr_timestamp_format [SQL Remote]

为从数据库检索得来的时间戳设置格式。

允许值

此格式字符串取自 `sr_date_format` 选项设置与 `sr_time_format` 选项设置的组合。

缺省值

`yyyy/mm/dd hh:nn:ss.Ssssss`

注释

消息代理使用此选项复制日期时间信息。缺省设置是 `sr_date_format` 选项设置与 `sr_time_format` 选项设置的组合。

另请参见

- “[sr_date_format 选项 \[SQL Remote\]](#)” 一节第 547 页
- “[sr_time_format 选项 \[SQL Remote\]](#)” 一节第 548 页
- “[SQL Remote 选项](#)” 一节 《SQL Remote》

string_rtruncation 选项 [兼容性]

确定当截断字符串时是否会出现错误。

允许值

On、Off

缺省值

On

注释

如果被截断的字符只包含空格，则不会出现异常。设置 On 与 ANSI/ISO SQL/2003 的行为对应。当此选项设置为 Off 时，不会出现异常，并且字符串会以静默方式被截断。

有几种情况可能会发生字符串截断。例如，如果声明的目标类型太短，则使用 INSERT、UPDATE、CAST 或向变量赋值可能会截断字符串。

另请参见

- “字符数据类型”一节 《SQL Anywhere 服务器 - SQL 参考》

subscribe_by_remote 选项 [SQL Remote]

控制对 NULL 或空字符串 SUBSCRIBE BY 值的解释。

允许值

On、Off

缺省值

On

注释

当此选项设置为 On 时，如果行具有的 SUBSCRIBE BY 值是 NULL 或空字符串，远程数据库在对这样的行进行操作时，会假定该远程用户预订了该行。当它被设置为 Off 时，会假定该远程用户没有预订此行。

此选项的唯一局限是：如果远程用户确实想要使用 NULL（空）或空预订表达式 INSERT（插入）或 UPDATE（更新）一行（仅针对在统一数据库保存的信息），则可能导致错误发生。这确实容易混淆，但可以通过在属于非远程用户的安装中分配预订值来避免这一情况。

另请参见

- “SQL Remote 选项”一节 《SQL Remote》
- “对多对多关系使用 subscribe_by_remote 选项”一节 《SQL Remote》

subsume_row_locks 选项 [数据库]

控制数据库服务器获取表个别行的锁的时间。

允许值

On、Off

缺省值

On

注释

如果 `subsume_row_locks` 选项为 `On` (缺省值), 则每当使用 `LOCK TABLE t IN EXCLUSIVE MODE` 以独占方式锁定表 `t` 时, 数据库服务器将不会再获得 `t` 的各个行的锁。

在单个事务中对 `t` 进行大范围更新时, 这会显著提高性能, 尤其是在 `t` 相对于高速缓存较大时。它还支持比锁定表当前所能处理的数量 (约 2-4 百万行) 更多的原子更新操作。

当此选项设置为 `On` 时, 以此方式锁定的表上的键集游标会在该数据库中的任何行修改后, 针对游标中的每一行返回行更改警告。注意, 其结果是, 数据库服务器可能将带 `ORDER BY` 的可更新游标转变为键集游标。

suppress_tds_debugging 选项 [数据库]

确定 TDS 调试信息是否出现在数据库服务器消息窗口中。

允许值

`On`、`Off`

缺省值

`Off`

注释

在以 `-z` 选项启动数据库服务器时, 调试信息会出现在数据库服务器消息窗口中, 其中包括有关 TDS 协议的调试信息。

`suppress_tds_debugging` 选项会限制出现在数据库服务器消息窗口中的有关 TDS 的调试信息。当此选项设置为 `Off` (缺省值) 时, TDS 调试信息会出现在数据库服务器消息窗口中。

synchronize_mirror_on_commit 选项 [数据库]

控制在异步或异步满页模式下运行时, 保证何时将数据库更改发送到镜像服务器。

允许值

`On`、`Off`

缺省值

`Off`

注释

`synchronize_mirror_on_commit` 选项允许精细控制在异步或异步满页模式下运行时保证何时将数据库更改发送到镜像服务器。缺省情况下, 此选项为 `Off`。此选项设置为 `On` 时, 每个 `COMMIT` 都会导致将事务日志中记录的所有更改发送到镜像服务器, 并致使镜像服务器在收到更改后向主服务器发送确认。可以使用 `SET TEMPORARY OPTION` 为特定事务设置此选项。通过在登录过程中检查 `APPINFO` 字符串为特定应用程序设置此选项也很有用。这样就允许定制镜像行为, 以满足不同应用程序的需要。

另请参见

- “数据库镜像简介”一节第 880 页

tds_empty_string_is_null 选项 [数据库]

控制为 TDS 连接将空字符串返回为 NULL 还是返回为包含一个空白字符的字符串。

允许值

On、Off

缺省值

Off

注释

缺省情况下，此选项被设置为 Off，并且 TDS 连接会将空字符串作为包含一个空白字符的字符串返回。当此选项设置为 On 时，TDS 连接会将空字符串作为 NULL 字符串返回。非 TDS 连接区分空字符串与 NULL 字符串。

temp_space_limit_check 选项 [数据库]

检查连接使用的临时文件空间量，当请求的空间量大于连接允许使用的限额时，请求将失败。

允许值

On、Off

缺省值

On

作用域

只能为 PUBLIC 组设置。需要 DBA 权限。

注释

当 temp_space_limit_check 设置为 On（缺省值）时，如果连接请求的临时文件空间量超出其限额，请求将失败并返回 SQLSTATE_TEMP_SPACE_LIMIT 错误。当此选项设置为 Off 时，数据库服务器不检查连接使用的临时文件空间量。当此选项设置为 Off 时，如果连接请求的临时空间量超出其限额，则将发生致命错误。

连接的临时文件空间限额为以下两个阈值中的最小值：

1. 如 max_temp_space 选项设置所指定的每个连接允许的最大临时文件空间量
2. 除以连接数后临时文件的最大可能大小

只有在临时文件增加到其最大大小的 80% 或更多时（这由操作系统所报告的设备上的剩余可用空间量决定）才使用此阈值。当连接请求的临时文件空间超出允许的限额时，该连接的当前请求失败，并产生错误 SQLSTATE 54W05 (TEMP_SPACE_LIMIT)。

可使用 `max_temp_space` 选项为连接使用的临时文件空间量指定一个硬性限制。

另请参见

- “`sa_disk_free_space` 系统过程” 一节 《SQL Anywhere 服务器 - SQL 参考》
- “`max_temp_space` 选项 [数据库]” 一节第 521 页

time_format 选项 [兼容性]

为从数据库中检索得到的时间设置格式。

允许值

由下面所列的符号组成的字符串

缺省值

`HH:NN:SS.sss`

作用域

可以为单个连接或 PUBLIC 组设置此选项。设置立即生效。

注释

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
<i>NN</i>	两位数表示的分钟数
<i>SS.ssssss</i>	秒及秒的小数部分，共六个小数位。并非所有平台都支持精度达到六位的时间戳。
<i>AA</i>	上午或下午（12 小时时钟）—对 24 小时时间省略 <i>AA</i> 和 <i>PP</i>
<i>PP</i>	如果需要则为下午（12 小时时钟）—对 24 小时时间省略 <i>AA</i> 和 <i>PP</i>

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也显示为大写。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。

另请参见

- “[date_format 选项 \[数据库\]](#)” 一节第 496 页
- “[timestamp_format 选项 \[兼容性\]](#)” 一节第 554 页

time_zone_adjustment 选项 [数据库]

允许修改连接的时区调整值。

允许值

整数（例如，300）

括在引号内的负整数（例如 '-300'）

以小时和分钟表示时间的字符串（跟在 + 或 - 后面并括在引号内，例如 '+5:00' 或 '-5:00'）

缺省值

如果客户端通过嵌入式 SQL、ODBC、OLE DB、ADO 或 ADO.NET 连接，则缺省值会根据客户端的时区来设置。如果客户端通过 jConnect 或 Open Client 连接，则缺省值会根据数据库服务器的时区来设置。

注释

time_zone_adjustment 选项值与 SELECT CONNECTION_PROPERTY('TimeZoneAdjustment'); 所返回的值相同。该值表示为了将时间显示为连接的本地时间而必须将协调通用时间（Coordinated Universal Time，简称 UTC）增加的分钟数。

另请参见

- TimeZoneAdjustment 属性：[“连接属性”](#) 一节第 566 页

timestamp_format 选项 [兼容性]

为从数据库检索得来的时间戳设置格式。

允许值

由下面所列的符号组成的字符串

缺省值

YYYY-MM-DD HH:NN:SS.SSS

作用域

可以为单个连接或 PUBLIC 组设置此选项。设置立即生效。

注释

格式为使用以下符号的字符串：

符号	说明
<i>YY</i>	两位年份
<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> ' 这样的格式），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
<i>DD</i>	以两位数表示的月中的某天
<i>DDD</i> [<i>d...</i>]	周中某天的字符简写形式
<i>HH</i>	两位数表示的小时数
<i>NN</i>	两位数表示的分钟数
<i>SS.ssssss</i>	秒及秒的小数部分，共六个小数位。并非所有平台都支持精度达到六位的时间戳。
<i>AA</i>	上午或下午（12 小时时钟）—对 24 小时时间省略 <i>AA</i> 和 <i>PP</i>
<i>PP</i>	如果需要则为下午（12 小时时钟）—对 24 小时时间省略 <i>AA</i> 和 <i>PP</i>

每个符号都会替换为按此设置格式的相应日期值。

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，可让 SQL Anywhere 为所用语言选择适当的大小写。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。

如果字符数据为多字节，则每个符号的长度反映字符数而不是字节数。例如，'*mmm*' 符号指定月份的长度为三个字符。

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。

注意

如果以重新排序日期格式的方式更改 `timestamp_format` 的设置，请确保更改 `date_order` 选项以使其反映相同的更改，反之亦然。请参见“[date_order 选项 \[数据库\]](#)”一节第 497 页。

另请参见

- “[date_format 选项 \[数据库\]](#)”一节第 496 页
- “[time_format 选项 \[兼容性\]](#)”一节第 553 页

truncate_timestamp_values 选项 [数据库] [MobiLink 客户端]

限制时间戳值的精度。

允许值

On、Off

缺省值

Off

作用域

只能为 PUBLIC 组设置。需要 DBA 权限。不应为已包含时间戳数据的数据库启用此选项。

注释

在 SQL Anywhere 中，TIMESTAMP 的值精确到六个小数位。然而，为保持与其它软件（可能会将 TIMESTAMP 值截断为三个小数位）的兼容性，您可以将 `truncate_timestamp_values` 选项设置为 On，以限制 SQL Anywhere 存储的小数位数。`default_timestamp_increment` 选项确定将 TIMESTAMP 的值截断为几位小数。

对于 MobiLink 同步，如果要设置此选项，则必须在执行第一次同步之前进行设置。

如果数据库服务器发现 TIMESTAMP 值的精度比 `truncate_timestamp_values` 和 `default_timestamp_increment` 的组合所指定的精度高，则会报告一个错误。

大多数情况下，确保使用正确的 TIMESTAMP 值的最简单方法是，卸载数据库然后重新将其装载到已设置 `truncate_timestamp_values` 和 `default_timestamp_increment` 值的新数据库中。不过，根据表中 TIMESTAMP 列的类型，也可以进行以下操作：

- 如果 TIMESTAMP 列用 DEFAULT TIMESTAMP 或 DEFAULT UTC TIMESTAMP 来定义（这样，修改行时将由数据库服务器自动更新该值），则必须先删除表中的所有行，然后再更改 `truncate_timestamp_values` 选项。您可以使用 DELETE 或 TRUNCATE TABLE 语句来删除行。
- 如果 TIMESTAMP 列用 DEFAULT TIMESTAMP 和 DEFAULT UTC TIMESTAMP 之外的其它值来定义，则可以执行将值转换为字符串然后又转换回 TIMESTAMP 的 UPDATE 语句。例如，

```
UPDATE T
  SET ts = CAST( DATEFORMAT( ts, 'yyyy/mm/dd hh:nn:ss.ss' )
    AS TIMESTAMP );
```

注意，此过程可能会损失过多的精度。使用的格式字符串取决于要保留的精度的位数。

另请参见

- “[default_timestamp_increment 选项 \[数据库\] \[MobiLink 客户端\]](#)” 一节第 500 页

示例

将 `default_timestamp_increment` 选项设置为 100000 会导致在秒部分中的第一个小数位后进行截断，从而会存储一个像 '2000/12/05 10:50:53:700' 这样的值。

tsql_outer_joins 选项 [兼容性]

控制在语句和视图中使用 Transact-SQL 外连接运算符 `*=` 和 `=*` 的能力。

允许值

On、Off

缺省值

Off

注释

不建议使用 Transact-SQL 外连接。将此选项设置为 On 允许您使用 Transact-SQL 外连接。

tsql_variables 选项 [兼容性]

控制 `@` 符号是否可以用作嵌入式 SQL 主机变量名的前缀。

允许值

On、Off

缺省值

Off

对 Open Client 和 jConnect 连接为 On

注释

当此选项设置为 On 时，可以使用 `@` 符号代替冒号作为嵌入式 SQL 中主机变量名的前缀。这主要是为与 Transact-SQL 兼容而执行的。

updatable_statement_isolation 选项 [数据库]

指定当 `isolation_level` 选项设置为只读语句快照时，可更新语句使用的隔离级别。

允许值

0, 1, 2, 3

缺省值

0

注释

当 `isolation_level` 选项设置为只读语句快照时，可更新语句使用 `updatable_statement_isolation` 选项指定的隔离级别。以下是接受的值：

- **0** 允许脏读、非重复性读取和幻像行。
- **1** 阻止进行脏读。允许非重复读取和幻像行。
- **2** 阻止进行脏读和非重复读取。允许幻像行。
- **3** 可串行读。阻止脏读、非重复性读取和幻像行。

另请参见

- [“isolation_level 选项 \[数据库\] \[兼容性\]”一节第 508 页](#)
- [“快照隔离”一节 《SQL Anywhere 服务器 - SQL 的用法》](#)
- [“隔离级别和一致性”一节 《SQL Anywhere 服务器 - SQL 的用法》](#)
- [“选择隔离级别”一节 《SQL Anywhere 服务器 - SQL 的用法》](#)

update_statistics 选项 [数据库]

控制执行查询期间统计信息的收集。

允许值

On、Off

缺省值

On

注释

数据库服务器会在查询正常执行期间收集统计信息，并使用收集到的统计信息自行调整列统计信息。可以将 `update_statistics` 选项设置为 Off，以在执行查询期间禁用统计信息收集功能。

正常情况下，无需关闭此选项。

`update_statistics` 选项不会影响因更新数据而导致的统计信息更改 (LOAD/INSERT/UPDATE/DELETE)。要控制统计信息是否根据这些语句而更新，请使用 `collect_statistics_on_dml_updates` 数据库选项。

`collect_statistics_on_dml_updates` 选项与 `update_statistics` 选项之间的差别在于：`update_statistics` 选项将满足某个谓语的实际行动数与预估满足该谓语的行数进行比较，然后更新估计值。

`collect_statistics_on_dml_updates` 选项根据插入、更新或删除的特定行的值修改列统计信息。

另请参见

- [“collect_statistics_on_dml_updates 选项 \[数据库\]”](#) 一节第 490 页
- [“更新列统计信息以提高优化程序性能”](#) 一节 《SQL Anywhere 服务器 - SQL 的用法》

user_estimates 选项 [数据库]

控制查询优化程序是考虑还是忽略查询谓语句中的用户选择性估计。

允许值

Enabled、Disabled、Override-Magic

缺省值

Override-Magic

作用域

可以为单个连接或 PUBLIC 组设置此选项。设置立即生效。

注释

SQL Anywhere 允许您指定用户选择性估计，这样，在数据库服务器无法准确预测谓语句的选择性时，可以提高优化程序的性能。然而，只能在适当的情况下应用用户选择性估计。例如，如果优化程序使用的 Override-Magic 选择性估计与实际选择性明显不同，则为涉及一个或多个函数的谓语句提供选择性估计可能很有用。

如果由于软件所选的访问计划较差，您为了解决由此造成的性能低下问题而使用了不准确的选择性估计，则建议您将此选项设置为 Disabled。如果您使用不准确的估计，数据库服务器可能不会选择一个最佳计划。

有关用户选择性估计的详细信息，请参见 [“显式选择性估计”](#) 一节 《SQL Anywhere 服务器 - SQL 参考》。

当用户选择性估计随谓语句一起提供时，会基于该选项的设置而考虑或忽略该估计。以下是接受的值：

- **Enabled** 考虑所有由用户提供的选择性估计。也可以使用 On 开启此选项。
- **Override-Magic** 仅在如下情况下使用用户选择性估计：如不使用，优化程序将只好选择采取最后的手段 - 启发值（也称魔术值）。
- **Disabled** 忽略所有用户估计，并在没有其它可用的估计数据时使用魔术值。也可以使用 Off 关闭此选项。

通过在 INSERT、UPDATE、DELETE、SELECT、UNION、EXCEPT 和 INTERSECT 语句中包括 OPTION 子句，可以替代此选项在各个语句中的任何临时或公共设置。请参见：

- “INSERT 语句”一节 《SQL Anywhere 服务器 - SQL 参考》
- “UPDATE 语句”一节 《SQL Anywhere 服务器 - SQL 参考》
- “DELETE 语句”一节 《SQL Anywhere 服务器 - SQL 参考》
- “SELECT 语句”一节 《SQL Anywhere 服务器 - SQL 参考》
- “UNION 子句”一节 《SQL Anywhere 服务器 - SQL 参考》
- “EXCEPT 子句”一节 《SQL Anywhere 服务器 - SQL 参考》
- “INTERSECT 子句”一节 《SQL Anywhere 服务器 - SQL 参考》

verify_all_columns 选项 [SQL Remote]

控制含有本地数据库发布的更新内容的消息是否与所有列值一起发送。

允许值

On、Off

缺省值

Off

注释

此选项只由 SQL Remote 使用。当它设置为 On 时，含有由本地数据库发布的更新内容的消息与所有列值一起发送，并且任意列中的冲突都会在预订者数据库中触发 RESOLVE UPDATE 触发器。

示例

以下语句在 SQL Anywhere 中针对所有用户将 verify_all_columns 选项设置为 Off:

```
SET OPTION PUBLIC.verify_all_columns = 'Off';
```

另请参见

- “SQL Remote 选项”一节 《SQL Remote》

verify_password_function 选项 [数据库]

使用 verify_password_function 选项来实现口令规则。

允许值

字符串

缺省值

空字符串（设置口令时不调用任何函数）。

作用域

需要 DBA 权限。

注释

创建或设置不为 NULL 的口令时，将自动调用由 `verify_password_function` 指定的函数。要防止用户替换函数，请将选项值设置为 `owner.function-name`。用户必须拥有口令才能连接到数据库。口令区分大小写且不可以：

- 以空格、单引号或双引号开头
- 以空格结尾
- 含有分号
- 长度超过 255 个字节

创建或更改口令时，口令先被转换为 UTF-8，然后再散列并存储在数据库中。如果数据库被卸载和重装到带有不同字符集的数据库，则现有口令将继续有效。如果数据库服务器不能将客户端的字符集转换为 UTF-8，建议采用由 7 位 ASCII 字符组成的口令，因为使用其它字符可能会无法正常工作。

可以使用以下任一语句来设置口令：

- CREATE USER
- ALTER USER
- GRANT

对用于创建或设置口令的语句进行校验之后，将调用函数，使用指定的规则来验证口令。如果口令符合指定的规则，该函数必须返回 NULL 以表示成功并执行调用的语句。否则，通过设置错误或返回非 NULL 字符串来指示出错。如果返回非 NULL 字符串，则该字符串将作为失败原因包括在错误中，以提供给用户。

口令验证函数具有两个参数：`user-name` VARCHAR(128) 和 `new-pwd` VARCHAR(255)。它返回类型为 VARCHAR(255) 的值。建议您对口令验证函数执行 ALTER FUNCTION `function-name` SET HIDDEN 语句，以确保它不能通过使用调试程序而分步执行。如果设置了 `verify_password_function` 选项，则不允许使用 GRANT CONNECT 语句指定一个以上的用户 ID 和口令。

有关口令规则的详细信息，请参见“使用口令验证”一节第 1007 页。

另请参见

- “`min_password_length` 选项 [数据库]”一节第 522 页
- “CREATE USER 语句”一节 《SQL Anywhere 服务器 - SQL 参考》
- “CREATE FUNCTION 语句 (Web 服务)”一节 《SQL Anywhere 服务器 - SQL 参考》
- “ALTER FUNCTION 语句”一节 《SQL Anywhere 服务器 - SQL 参考》
- “ALTER USER 语句”一节 《SQL Anywhere 服务器 - SQL 参考》
- “GRANT 语句”一节 《SQL Anywhere 服务器 - SQL 参考》
- “提高口令安全性”一节第 1006 页
- “NewPassword 连接参数 [NEWPWD]”一节第 274 页

示例

以下示例定义表和函数并设置一些登录策略选项。它们共同实现了高级口令规则，其中包括要求在口令中使用某些类型的字符、不允许口令重用和为口令设置有效期。当创建用户 ID 或更改口令时，数据库服务器会通过 `verify_password_function` 选项调用函数。应用程序可以调用 `post_login_procedure` 选项指定的过程，以报告口令应在到期前进行更改。

也可以从以下位置获得此示例的代码：`samples-dir\SQLAnywhere\SQL\verify_password.sql`。（有关 `samples-dir` 的信息，请参见“[示例目录](#)”一节第 368 页。）

```
-- only DBA should have permissions on this table
CREATE TABLE DBA.t_pwd_history(
    pk          INT          DEFAULT AUTOINCREMENT PRIMARY KEY,
    user_name   CHAR(128),   -- the user whose password is set
    pwd_hash    CHAR(32) );  -- hash of password value to detect
                           -- duplicate passwords

-- called whenever a non-NULL password is set
-- to verify the password conforms to password rules
CREATE FUNCTION DBA.f_verify_pwd( uid      VARCHAR(128),
                                  new_pwd  VARCHAR(255) )
RETURNS VARCHAR(255)
BEGIN
    -- a table with one row per character in new_pwd
    DECLARE local temporary table pwd_chars(
        pos INT PRIMARY KEY,    -- index of c in new_pwd
        c   CHAR( 1 CHAR ) );  -- character
    -- new_pwd with non-alpha characters removed
    DECLARE pwd_alpha_only    CHAR(255);
    DECLARE num_lower_chars   INT;

    -- enforce minimum length (can also be done with
    -- min_password_length option)
    IF length( new_pwd ) < 6 THEN
        RETURN 'password must be at least 6 characters long';
    END IF;

    -- break new_pwd into one row per character
    INSERT INTO pwd_chars SELECT row_num, substr( new_pwd, row_num, 1 )
        FROM dbo.RowGenerator
        WHERE row_num <= length( new_pwd );

    -- copy of new_pwd containing alpha-only characters
    SELECT list( c, '' ORDER BY pos ) INTO pwd_alpha_only
        FROM pwd_chars WHERE c BETWEEN 'a' AND 'z' OR c BETWEEN 'A' AND 'Z';

    -- number of lower case characters IN new_pwd
    SELECT count(*) INTO num_lower_chars
        FROM pwd_chars WHERE CAST( c AS BINARY ) BETWEEN 'a' AND 'z';

    -- enforce rules based on characters contained in new_pwd
    IF ( SELECT count(*) FROM pwd_chars WHERE c BETWEEN '0' AND '9' )
        < 1 THEN
        RETURN 'password must contain at least one numeric digit';
    ELSEIF length( pwd_alpha_only ) < 2 THEN
        RETURN 'password must contain at least two letters';
    ELSEIF num_lower_chars = 0
        OR length( pwd_alpha_only ) - num_lower_chars = 0 THEN
        RETURN 'password must contain both upper- and lowercase characters';
    END IF;

    -- not the same as any user name
```

```

-- (this could be modified to check against a disallowed words table)
IF EXISTS( SELECT * FROM SYS.SYSUSER
           WHERE lower( user_name ) IN ( lower( pwd_alpha_only ),
                                         lower( new_pwd ) ) ) THEN
    RETURN 'password or only alphabetic characters in password ' ||
           'must not match any user name';
END IF;

-- not the same as any previous password for this user
IF EXISTS( SELECT * FROM t_pwd_history
           WHERE user_name = uid
             AND pwd_hash = hash( uid || new_pwd, 'md5' ) ) THEN
    RETURN 'previous passwords cannot be reused';
END IF;

-- save the new password
INSERT INTO t_pwd_history( user_name, pwd_hash )
VALUES( uid, hash( uid || new_pwd, 'md5' ) );

RETURN( NULL );
END;

ALTER FUNCTION DBA.f_verify_pwd SET HIDDEN;
GRANT EXECUTE ON DBA.f_verify_pwd TO PUBLIC;
SET OPTION PUBLIC.verify_password_function = 'DBA.f_verify_pwd';

-- All passwords expire in 180 days. Expired passwords can be changed
-- by the user using the NewPassword connection parameter.
ALTER LOGIN POLICY DEFAULT password_life_time = 180;

-- If an application calls the procedure specified by the
-- post_login_procedure option, then the procedure can be used to
-- warn the user that their password is about to expire. In particular,
-- Interactive SQL and Sybase Central call the post_login_procedure.
ALTER LOGIN POLICY DEFAULT password_grace_time = 30;

-- Five consecutive failed login attempts will result in a non-DBA
-- user ID being locked.
ALTER LOGIN POLICY DEFAULT max_failed_login_attempts = 5;

```

verify_threshold 选项 [SQL Remote]

控制在复制更新内容时验证的列。

允许值

以字节为单位的整数

缺省值

1000

注释

此选项只由 SQL Remote 使用。如果列的数据类型的长度超过阈值，则在复制 UPDATE 时不验证旧的列值。这样可以控制住 SQL Remote 消息的大小，但弊端是检测不到冲突的长值更新。

另请参见

- “SQL Remote 选项”一节 《SQL Remote》

wait_for_commit 选项 [数据库]

判断在处理数据时检查外键完整性的时间。

允许值

On、Off

缺省值

Off

作用域

可以为单个连接或 PUBLIC 组设置此选项。设置立即生效。

注释

如果此选项设置为 On，则在下一条 COMMIT 语句之前，数据库并不检查外键完整性。否则，在插入、更新或删除所有不是使用 check_on_commit 选项创建的外键时，都会对其进行检查。

webservice_namespace_host 选项 [数据库]

指定在生成的 WSDL 文档中用作 XML 命名空间的主机名。

允许值

NULL 或主机名字符串

缺省值

NULL

作用域

只能为 PUBLIC 组设置。设置立即生效。需要 DBA 权限。

注释

Web 服务描述语言文档 (WSDL) 由 DISH 服务导出。这些是包含可用 SOAP 服务说明的 XML 文档。此 XML 文档中的 targetNamespace 和 soapAction 操作的 URL 包含一个主机名。当此选项设置为 NULL (缺省值) 时，主机名是运行数据库服务器的计算机的名称。如果此选项设置为一个字符串值，则该字符串将用作主机名。有些 Web 服务客户端应用程序在部署后，不再以开发该应用程序的主机为目标，此选项正适用于在开发这样的 Web 服务客户端应用程序时使用。

连接属性、数据库属性和数据库服务器属性

目录

连接属性	566
数据库服务器属性	588
数据库属性	601

连接属性

下面表格列出了 SQL Anywhere 数据库的每个连接的可用属性。

可以使用 CONNECTION_PROPERTY 系统函数检索单个属性的值，也可使用 sa_conn_properties 系统过程检索所有连接属性的值。属性名称不区分大小写。

示例

◆ 检索单个连接属性的值：

- 使用 CONNECTION_PROPERTY 系统函数。以下语句返回已通过当前连接从文件中读取的页数。

```
SELECT CONNECTION_PROPERTY ( 'DiskRead' );
```

◆ 检索所有连接属性的值：

- 使用 sa_conn_properties 系统过程：

```
CALL sa_conn_properties( );
```

每个连接属性单独占一行。

另请参见

- “CONNECTION_PROPERTY 函数 [System]” 一节 《SQL Anywhere 服务器 - SQL 参考》
- “sa_conn_activity 系统过程” 一节 《SQL Anywhere 服务器 - SQL 参考》
- “数据库服务器属性” 一节第 588 页
- “数据库属性” 一节第 601 页

说明

属性	说明
allow_nulls_by_default	返回一个值，指示是否允许创建时未指定 NULL 或 NOT NULL 的列中包含空值。请参见 “ allow_nulls_by_default 选项 [兼容性] ” 一节第 477 页。
allow_read_client_file	返回一个值，指示数据库服务器是否允许读取客户端计算机上的文件。请参见 “ allow_read_client_file 选项 [数据库] ” 一节第 478 页。
allow_snapshot_isolation	返回一个值，指示是启用还是禁用快照隔离。请参见 “ allow_snapshot_isolation 选项 [数据库] ” 一节第 478 页。
allow_write_client_file	返回一个值，指示数据库服务器是否允许向客户端计算机写入文件。请参见 “ allow_write_client_file 选项 [数据库] ” 一节第 479 页。

属性	说明
ansi_blanks	返回一个值，指示何时在客户端截断字符数据。请参见“ ansi_blanks 选项 [兼容性] ”一节第 480 页。
ansi_close_cursors_on_rollback	返回一个值，指示是否在执行 ROLLBACK 时关闭通过 WITH HOLD 打开的游标。请参见“ ansi_close_cursors_on_rollback 选项 [兼容性] ”一节第 481 页。
ansi_permissions	返回一个值，指示是否针对 DELETE 和 UPDATE 语句检查权限。请参见“ ansi_permissions 选项 [兼容性] ”一节第 481 页。
ansi_substring	返回一个值，指示在为起始参数或长度参数提供了负值时 SUBSTRING (SUBSTR) 函数的行为。请参见“ ansi_substring 选项 [兼容性] ”一节第 482 页。
ansi_update_constraints	返回一个值，指示允许的更新范围。请参见“ ansi_update_constraints 选项 [兼容性] ”一节第 483 页。
ansinull	返回一个值，指示如何解释空值。请参见“ ansinull 选项 [兼容性] ”一节第 483 页。
AppInfo	<p>返回有关建立连接的客户端的信息。对于 HTTP 连接，这包括有关浏览器的信息。对于使用较旧版本 jConnect 或 Open Client 的连接，信息可能不完整。</p> <p>API 值可为 DBLIB、ODBC、OLEDB、ADO.NET、iAnywhereJDBC、PHP、PerlDBD 或 DBEXPRESS。</p> <p>有关其它类型连接返回值的详细信息，请参见“AppInfo 连接参数 [APP]”一节第 249 页。</p>
ApproximateCPUTime	返回由给定连接累计的 CPU 时间量的估计值，单位为秒。该值与实际值的最大差异可达 50%，不过返回值的典型变动范围是 5-10%。在多处理器计算机上，每个 CPU（或是超线程或内核）都会累计时间，因此所有连接的累计时间总和可能大于所经过的时间。在 Windows 和 Linux 上支持此属性。
auditing	<p>如果将 PUBLIC.auditing 选项设置为 On，则返回 On。否则，返回 Off。</p> <p>如果将 auditing 选项设置为 On，而将 conn_auditing 选项设置为 Off，则即使未对当前连接进行审计，auditing 连接属性仍将返回 On。请参见“控制审计”一节第 1012 页和“auditing 选项 [数据库]”一节第 484 页。</p>

属性	说明
auditing_options	此属性保留供系统使用。请不要更改此选项的设置。
Authenticated	如果应用程序发送了有效的连接验证字符串，则返回 Yes。如果应用程序尚未发送有效的连接验证字符串，则返回 No。
AuthType	返回连接时所使用的验证类型。返回值可能是 Standard、Integrated、Kerberos 或空字符串。对于内部连接以及使用 AUTHORIZATION OFF 的 HTTP 服务连接，将返回空字符串。
background_priority	不建议使用。返回一个值，指示当前连接对其它连接的性能的影响程度。请参见“ background_priority 选项 [数据库] [不建议使用] ”一节第 485 页。
BlockedOn	如果当前连接未被阻塞，则返回零；如果当前连接被阻塞，则返回由于锁定冲突而阻塞连接的连接号。
blocking	返回一个值，指示数据库服务器对锁定冲突的响应行为。请参见“ blocking 选项 [数据库] ”一节第 486 页。
blocking_timeout	返回事务为获得锁而等待的时间长度（以毫秒为单位）。请参见“ blocking_timeout 选项 [数据库] ”一节第 487 页。
BytesReceived	返回在客户端/服务器通信期间所收到的字节数。此值对于 HTTP 和 HTTPS 连接会进行更新。
BytesReceivedUncomp	返回在禁用压缩的情况下，在客户端/服务器通信期间所收到的字节数。在禁用压缩的情况下，此值与 BytesReceived 的值相同。
BytesSent	返回在客户端/服务器通信期间所发送的字节数。此值对于 HTTP 和 HTTPS 连接会进行更新。
BytesSentUncomp	返回在禁用压缩的情况下，在客户端/服务器通信期间将发送的字节数。在禁用压缩的情况下，此值与 BytesSent 的值相同。
CacheHits	返回成功读取高速缓存的次数。
CacheRead	返回已在高速缓存中查找的数据库页数。
CacheReadIndInt	返回已从高速缓存中读取的索引内部节点页数。
CacheReadIndLeaf	返回已从高速缓存中读取的索引叶页数。

属性	说明
CacheReadTable	返回已从高速缓存中读取的表页数。
CacheReadWorkTable	返回高速缓存工作表读取次数。
CarverHeapPages	返回供短期使用（如查询优化）的堆页数。
chained	返回在缺少 BEGIN TRANSACTION 语句时使用的事务模式。请参见“ chained 选项 [兼容性] ”一节第 487 页。
CharSet	返回连接所使用的 CHAR 字符集。
checkpoint_time	返回数据库服务器在不执行检查点操作的情况下运行的最长时间（以分钟为单位）。请参见“ checkpoint_time 选项 [数据库] ”一节第 488 页。
cis_option	如果远程数据访问的调试信息出现在数据库服务器消息窗口中，则返回 0；如果远程数据访问的调试信息不出现在数据库服务器消息窗口中，则返回 7。请参见“ cis_option 选项 [数据库] ”一节第 488 页。
cis_rowset_size	返回每次读取从远程服务器返回的行数。请参见“ cis_rowset_size 选项 [数据库] ”一节第 489 页。
ClientLibrary	对于 jConnect 连接，返回 jConnect；对于 Open Client 连接，返回 CT_Library；对于 HTTP 连接，返回 None；对于 ODBC、嵌入式 SQL、OLE DB、ADO.NET 和 iAnywhere JDBC 驱动程序连接，返回 CmdSeq。
ClientNodeAddress	返回客户端/服务器连接中客户端的节点。当客户端和服务端均在同一台计算机上时，将会返回一个空字符串。这是 NodeAddress 属性的同义词。
ClientPort	返回客户端的 TCP/IP 端口号；或者如果连接不是 TCP/IP 连接，则返回 0。
ClientStmtCacheHits	返回因客户端语句高速缓存而不需要对此连接执行的准备的数目。如果禁用了客户端语句高速缓存，则为必需的附加准备数。请参见“ max_client_statements_cached 选项 [数据库] ”一节第 516 页。
ClientStmtCacheMisses	返回客户端语句高速缓存中为此连接再次准备的语句的数目。即为由于模式更改、数据库选项设置或 DROP VARIABLE 语句等原因而使本打算重用的某个高速缓存的语句无法重用的次数。请参见“ max_client_statements_cached 选项 [数据库] ”一节第 516 页。

属性	说明
close_on_endtrans	返回 On 或 Off 以指示是否在事务结束时关闭游标。请参见“ close_on_endtrans 选项 [兼容性] ”一节第 489 页。
collect_statistics_on_dml_updates	返回 On 或 Off 以指示是否在执行数据修改 DML 语句（如 INSERT、DELETE 和 UPDATE）期间收集统计信息。请参见“ collect_statistics_on_dml_updates 选项 [数据库] ”一节第 490 页。
Commit	返回已处理的提交请求数。
CommLink	返回连接的通信链接。这是 SQL Anywhere 所支持的网络协议之一，或者 local，如果为相同计算机连接。
CommNetworkLink	返回连接的通信链接。这是 SQL Anywhere 所支持的网络协议之一。值包括 SharedMemory 和 TCPIP。 CommNetworkLink 属性总是返回链接的名称，不管它是否为相同计算机。
CommProtocol	对于 Open Client 和 jConnect 连接，返回 TDS；对于 HTTP 连接，返回 HTTP；对于 ODBC、嵌入式 SQL、OLE DB、ADO.NET 和 iAnywhere JDBC 驱动程序连接，返回 CmdSeq。
Compression	返回 On 或 Off 以指示是否对连接启用通信压缩。
conn_auditing	如果为连接启用了审计，则即使将 auditing 选项设置为 Off，也会返回 On。请参见“ 控制审计 ”一节第 1012 页。
connection_authentication	返回用于验证客户端的字符串。修改数据库前需要进行验证。请参见“ connection_authentication 选项 [数据库] ”一节第 491 页。
continue_after_raiserror	返回 On 或 Off 以指示每当遇到 RAISERROR 语句时，是否停止执行过程或触发器。请参见“ continue_after_raiserror 选项 [兼容性] ”一节第 492 页。
conversion_error	返回 On 或 Off 以指示在从数据库读取信息时报告数据类型转换失败。请参见“ conversion_error 选项 [兼容性] ”一节第 493 页。
cooperative_commit_timeout	返回数据库服务器在写入磁盘之前等待其它连接填充日志页的时间（以毫秒为单位）。请参见“ cooperative_commit_timeout 选项 [数据库] ”一节第 493 页。

属性	说明
cooperative_commits	返回 On 或 Off 以指示何时将提交写入磁盘。请参见“ cooperative_commits 选项 [数据库] ”一节第 494 页。
CurrentLineNumber	返回连接正在执行的过程或复合语句的当前行号。可使用 CurrentProcedure 属性标识该过程。如果该行是来自客户端的复合语句的一部分，则返回一个空字符串。
CurrentProcedure	返回连接当前正在执行的过程的名称。如果该连接正在执行嵌套过程调用，则该名称为当前过程的名称。如果未在执行任何过程，则返回一个空字符串。
Cursor	返回服务器当前维护的声明游标数。
CursorOpen	返回服务器当前维护的打开游标数。
database_authentication	返回用于验证数据库的字符串。需要先对已验证的数据库服务器进行验证，才能修改数据库。请参见“ database_authentication [数据库] ”一节第 495 页。
date_format	返回一个字符串，指示从数据库检索的日期格式。请参见“ date_format 选项 [数据库] ”一节第 496 页。
date_order	返回一个字符串，指示如何设置日期格式。请参见“ date_order 选项 [数据库] ”一节第 497 页。
DBNumber	返回数据库的 ID 编号。
debug_messages	返回 On 或 Off 以指示是否执行包含 DEBUG ONLY 子句的 MESSAGE 语句。请参见“ debug_messages 选项 [数据库] ”一节第 498 页。
dedicated_task	返回 On 或 Off 以指示请求处理任务是否专门用于处理连接请求。请参见“ dedicated_task 选项 [数据库] ”一节第 499 页。
default_dbpace	返回缺省 dbspace 的名称，如果尚未指定缺省的 dbspace，则返回一个空字符串。请参见“ default_dbpace 选项 [数据库] ”一节第 499 页。
default_timestamp_increment	返回一个值（以毫秒为单位），该值将被增加到类型为 TIMESTAMP 的列中以使该列中的值保持唯一。请参见“ default_timestamp_increment 选项 [数据库] [MobiLink 客户端] ”一节第 500 页。

属性	说明
delayed_commit_timeout	返回数据库服务器在执行 COMMIT 后等待将控制返回到应用程序的时间（以毫秒为单位）。请参见“ delayed_commit_timeout 选项 [数据库] ”一节第 501 页。
delayed_commits	返回 On 或 Off 以指示数据库服务器在执行 COMMIT 后何时将控制返回到应用程序。请参见“ delayed_commits 选项 [数据库] ”一节第 501 页。
DiskRead	返回已从磁盘读取的页数。
DiskReadHint	返回磁盘读取提示数。
DiskReadHintPages	返回磁盘读取提示页数。
DiskReadIndInt	返回已从磁盘读取的索引内部节点页数。
DiskReadIndLeaf	返回已从磁盘读取的索引叶页数。
DiskReadTable	返回已从磁盘读取的表页数。
DiskReadWorkTable	返回磁盘工作表读取数。
DiskSyncRead	返回同步发出的磁盘读取数。
DiskSyncWrite	返回同步发出的写入数。
DiskWaitRead	返回数据库服务器等待异步读取的次数。
DiskWaitWrite	返回数据库服务器等待异步写入的次数。
DiskWrite	返回已写入磁盘的修改页数。
DiskWriteHint	返回磁盘写入提示数。
DiskWriteHintPages	返回磁盘写入提示页数。
Encryption	返回一个值，指示是否对连接加密。请参见“ Encryption 连接参数 [ENC] ”一节第 265 页。
escape_character	此属性保留供系统使用。请不要更改此选项的设置。
EventName	如果连接正在运行事件处理程序，则返回相关事件的名称。否则，结果为 NULL。
exclude_operators	此属性保留供系统使用。请不要更改此选项的设置。

属性	说明
ExprCacheAbandons	返回表达式高速缓存因命中率太低而被放弃的次数。
ExprCacheDropsToReadOnly	返回表达式高速缓存因命中率低而降为只读状态的次数。
ExprCacheEvicts	返回从表达式高速缓存中的退出次数。
ExprCacheHits	返回表达式高速缓存中的命中次数。
ExprCacheInserts	返回在表达式高速缓存中插入的值的个数。
ExprCacheLookups	返回在表达式高速缓存中执行的查找次数。
ExprCacheResumesOfReadWrite	返回表达式高速缓存因命中率上升而恢复读写状态的次数。
ExprCacheStarts	返回表达式高速缓存的启动次数。
extended_join_syntax	如果允许使用具有多表连接的重复相关名语法的查询，则返回 On，如果将此类查询作为错误报告，则返回 Off。请参见“ extended_join_syntax 选项 [数据库] ”一节第 503 页。
fire_triggers	如果在数据库中触发触发器，则返回 On；否则，返回 Off。请参见“ fire_triggers 选项 [兼容性] ”一节第 504 页。
first_day_of_week	返回用于表示每周第一天的数字，其中 7=星期日，1=星期一。请参见“ first_day_of_week 选项 [数据库] ”一节第 504 页。
for_xml_null_treatment	当在查询中使用了 FOR XML 子句时，如果从结果中省略包含空值的元素和属性，则返回 Omit；如果为空值生成空元素或属性，则返回 Empty。请参见“ for_xml_null_treatment 选项 [数据库] ”一节第 505 页。
force_view_creation	此属性保留供系统使用。请不要更改此选项的设置。
FullCompare	返回已执行的超过索引中散列值的比较次数。
GetData	返回 GETDATA 请求数。
global_database_id	返回用于以 DEFAULT GLOBAL AUTOINCREMENT 创建的列的起始值。请参见“ global_database_id 选项 [数据库] ”一节第 506 页。
HashForcedPartitions	返回因内存争用而强制散列运算符进行分区的次数。
HashRowsFiltered	返回位矢量过滤器拒绝的探测行数。
HashRowsPartitioned	返回写入散列工作表的行数。

属性	说明
HashWorkTables	返回为基于散列的操作而创建的工作表数。
HeapsCarver	返回供短期使用（如查询优化）的堆数。
HeapsLocked	返回高速缓存中当前锁定的可重定位堆数。
HeapsQuery	返回用于查询处理（散列和排序操作）的堆数。
HeapsRelocatable	返回可重定位堆数。
http_session_timeout	返回当前 HTTP 会话超时值（以分钟为单位）。请参见“ http_session_timeout 选项 [数据库] ”一节第 507 页。
HttpServiceName	返回 Web 应用程序的服务名起源。该属性对于错误报告和控制流很有用。当从中选择此属性的存储过程并非源自于 HTTP 请求时，或者如果连接当前为等待继续进行 HTTP 会话而处于不活动状态，将会返回一个空字符串。
IdleTimeout	返回连接的空闲超时值。请参见“ Idle 连接参数 ”一节第 268 页。
IndAdd	返回已添加到索引的条目数。
IndLookup	返回已在索引中查找的条目数。
integrated_server_name	返回域控制器服务器的名称，该服务器用于查找集成登录的 Windows 用户组成员资格。请参见“ integrated_server_name 选项 [数据库] ”一节第 507 页。
IsDebugger	返回 Yes 或 No 以区分出正用于运行 SQL Anywhere 调试程序的连接。如果当前连接编号对应于某调试程序的连接编号，此值为 Yes，否则为 No。请参见“ 调试过程、函数、触发器和事件 ”《SQL Anywhere 服务器 - SQL 的用法》。
isolation_level	返回连接的隔离级别（0、1、2、3、快照、语句快照或只读语句快照）。请参见“ isolation_level 选项 [数据库] [兼容性] ”一节第 508 页。
java_location	返回数据库的 Java VM 路径（如果已指定）。请参见“ java_location 选项 [数据库] ”一节第 509 页。
java_main_userid	返回数据库用户的名称，可使用该用户的连接安装类以及执行其它与 Java 相关的管理任务。请参见“ java_main_userid 选项 [数据库] ”一节第 510 页。

属性	说明
java_vm_options	返回数据库服务器启动 Java VM 时使用的命令行选项。请参见“ java_vm_options 选项 [数据库] ”一节第 510 页。
Language	返回地区语言。
LastIdle	返回请求间隔时间。
LastPlanText	返回在连接上执行的最后一个查询的长文本计划。通过使用 <code>sa_server_option</code> 系统过程的 <code>RememberLastPlan</code> 选项或使用 <code>-zp</code> 服务器选项进行设置，可对最后一个计划的记忆情况进行控制。请参见“ -zp 服务器选项 ”一节第 230 页。
LastReqTime	返回指定连接的最后一个请求开始的时间。
LastStatement	<p>返回最近为当前连接准备的 SQL 语句。请参见“-zl 服务器选项”一节第 227 页。</p> <p>当准备了语句时，将会设置 <code>LastStatement</code> 值；当删除了语句时，将会清除该值。每个连接仅会记忆一个语句字符串。</p> <p>如果 <code>sa_conn_activity</code> 对某个连接报告了一个非空值，它很可能为该连接当前执行的语句。假如该语句已完成，则很有可能已丢弃了它并清除了属性值。如果某个应用程序准备了多条语句并保有其语句句柄，则 <code>LastStatement</code> 值不反映连接当前进行的操作。</p> <p>如果启用客户端语句高速缓存并且重用了某个高速缓存的语句，则此属性返回空字符串。</p>
LivenessTimeout	返回当前连接的活动超时期。请参见“ LivenessTimeout 连接参数 [LTO] ”一节第 272 页。
lock_rejected_rows	此属性保留供系统使用。请不要更改此选项的设置。
LockCount	返回连接所持有的锁的个数。
LockIndexID	返回锁定索引的标识符。
LockName	返回一个 64 位无符号整数值，表示连接正在等待的锁。
LockRowID	返回锁定行的标识符。

属性	说明
LockTableOID	如果连接未被阻塞或连接与调用 CONNECTION _PROPERTY 连接位于不同的数据库，则返回零。否则，此值是与此连接正在等待的锁相应的那个表的对象 ID。可使用该对象 ID 通过 SYSTAB 系统视图查找表信息。请参见“SYSTAB 系统视图”一节《SQL Anywhere 服务器 - SQL 参考》。
log_deadlocks	如果报告死锁信息，则返回 On；否则，返回 Off。请参见“log_deadlocks 选项 [数据库]”一节第 511 页。
LogFreeCommit	返回重做无偿提交数。当请求提交事务日志而该日志已被写入（从而提交是无偿完成的）时，便会发生重做无偿提交。
login_mode	返回 Standard、Integrated 或 Kerberos 中的一个或多个以指示是否支持集成登录和 Kerberos。请参见“login_mode 选项 [数据库]”一节第 512 页。
login_procedure	返回用于在启动时设置兼容性选项的存储过程的名称。请参见“login_procedure 选项 [数据库]”一节第 513 页。
LoginTime	返回建立连接的日期和时间。
LogWrite	返回已写入事务日志的页数。
materialized_view_optimization	<p>返回一个值，指示在查询优化期间是否使用实例化视图：</p> <ul style="list-style-type: none"> ● Disabled ● Fresh ● Stale ● N Minute[s] ● N Hour[s] ● N Day[s] ● N Week[s] ● N Month[s] <p>请参见“materialized_view_optimization 选项 [数据库]”一节第 515 页。</p>
max_client_statements_cached	返回由客户端高速缓存的语句数。请参见“max_client_statements_cached 选项 [数据库]”一节第 516 页。
max_cursor_count	返回一个值，指定连接一次最多可以使用的游标数。请参见“max_cursor_count 选项 [数据库]”一节第 517 页。

属性	说明
max_hash_size	不建议使用此属性。
max_plans_cached	返回一个值，指定最多要在高速缓存中存储的执行计划数。请参见“ max_plans_cached 选项 [数据库] ”一节第 517 页。
max_priority	返回一个值，指示连接可以具有的最高优先级级别。请参见“ max_priority 选项 [数据库] ”一节第 518 页。
max_query_tasks	返回数据库服务器在处理查询时最多可以使用的请求数。请参见“ max_query_tasks 选项 [数据库] ”一节第 519 页。
max_recursive_iterations	返回一个值，指定递归公用表表达式最多可以进行的迭代次数。请参见“ max_recursive_iterations 选项 [数据库] ”一节第 520 页。
max_statement_count	返回一个值，指定连接最多可以同时使用的经过准备的语句数。请参见“ max_statement_count 选项 [数据库] ”一节第 520 页。
max_temp_space	返回一个值，指示连接可用的最大临时文件空间量。请参见“ max_temp_space 选项 [数据库] ”一节第 521 页。
MessageReceived	返回由导致 WAITFOR 语句中断的 MESSAGE 语句所生成的字符串。否则，返回一个空字符串。
min_password_length	返回数据库中新口令的最小长度。请参见“ min_password_length 选项 [数据库] ”一节第 522 页。
Name	返回当前连接的名称。
NcharCharSet	返回连接所使用的 NCHAR 字符集。
nearest_century	返回一个值，指示在字符串到日期转换中如何对两位数年份进行解释。请参见“ nearest_century 选项 [兼容性] ”一节第 523 页。
NodeAddress	返回客户端/服务器连接中客户端的节点。当客户端和服务端均在同一台计算机上时，将会返回一个空字符串。
non_keywords	返回为使其能够用作标识符而关闭的关键字列表（如果有）。请参见“ non_keywords 选项 [兼容性] ”一节第 523 页。
Number	返回连接的 ID 编号。

属性	说明
odbc_describe_binary_as_varbinary	如果 SQL Anywhere ODBC 驱动程序将 BINARY 和 VARBINARY 列都描述为 SQL_BINARY，则返回 Off；如果该 ODBC 驱动程序将 BINARY 和 VARBINARY 列都描述为 SQL_VARBINARY，则返回 On。请参见“ odbc_describe_binary_as_varbinary [数据库] ”一节第 524 页。
odbc_distinguish_char_and_varchar	如果将 CHAR 列描述为 SQL_VARCHAR，则返回 Off；如果将 CHAR 列描述为 SQL_CHAR，则返回 On。请参见“ odbc_distinguish_char_and_varchar 选项 [数据库] ”一节第 525 页。
oem_string	返回数据库文件头页中存储的字符串。请参见“ oem_string 选项 [数据库] ”一节第 525 页。
on_charset_conversion_failure	返回 Ignore、Warning 或 Error 之一，以指示在字符集转换期间遇到错误时的行为。请参见“ on_charset_conversion_failure 选项 [数据库] ”一节第 527 页。
on_tsq_error	返回 Stop、Conditional 或 Continue 之一，以指示在执行存储过程期间遇到错误时的行为。请参见“ on_tsq_error 选项 [兼容性] ”一节第 528 页。
optimization_goal	返回 First-row 或 All-rows 之一，以指示如何对查询处理进行优化。请参见“ optimization_goal 选项 [数据库] ”一节第 528 页。
optimization_level	返回一个介于 0 到 15 的值。此数用于控制 SQL Anywhere 查询优化程序为查找 SQL 语句的访问计划而付出的工作量。请参见“ optimization_level 选项 [数据库] ”一节第 529 页。
optimization_workload	返回一个值，指示 SQL Anywhere 查询优化程序为查找 SQL 语句的访问计划而付出的工作量。请参见“ optimization_workload 选项 [数据库] ”一节第 530 页。
OSUser	返回与客户端进程相关联的操作系统用户名。如果客户端进程模拟另一个用户（或者设置的 ID 位在 Unix 上设置），则返回被模拟的用户名。对于 10.0.1 版及更早版本的客户端，以及 HTTP 和 TDS 客户端，将返回空字符串。
PacketSize	返回连接所使用的包大小（以字节为单位）。

属性	说明
PacketsReceived	返回所收到的客户端/服务器通信包数。此值对于 HTTP 和 HTTPS 连接不会进行更新。
PacketsReceivedUncomp	返回在禁用压缩的情况下，在客户端/服务器通信期间所收到的包数。（在禁用压缩的情况下，该值与 PacketsReceived 的值相同。）
PacketsSent	返回所发送的客户端/服务器通信包数。此值对于 HTTP 和 HTTPS 连接不会进行更新。
PacketsSentUncomp	返回在禁用压缩的情况下，在客户端/服务器通信期间所发送的包数。（在禁用压缩的情况下，该值与 PacketsSent 的值相同。）
pinned_cursor_percent_of_cache	返回可用于固定游标的高速缓存百分比。请参见“ pinned_cursor_percent_of_cache 选项 [数据库] ”一节第 531 页。
post_login_procedure	返回结果集中包含消息的过程名，这些消息应在用户连接时由应用程序显示。请参见“ post_login_procedure 选项 [数据库] ”一节第 531 页。
precision	返回数字位数和数值精度设置。请参见“ precision 选项 [数据库] ”一节第 533 页。
prefetch	如果不进行预取，返回 Off；如果进行预取（但游标类型为 SENSITIVE 或查询中包含代理表时则不进行预取），返回 Conditional；如果即使对于 SENSITIVE 游标类型和包含代理表的游标也进行预取，返回 Always。请参见“ prefetch 选项 [数据库] ”一节第 534 页。
Prepares	返回为进行连接所执行的语句准备次数。
PrepStmt	返回服务器当前维护的经过准备的语句数。
preserve_source_format	如果在系统表中保存过程、触发器、视图和事件处理程序的原始源代码，则返回 On；否则，返回 Off。请参见“ preserve_source_format 选项 [数据库] ”一节第 534 页。
prevent_article_pkey_update	如果不允许对发布中所涉及的表的主键列进行更新，则返回 On；否则，返回 Off。请参见“ prevent_article_pkey_update 选项 [数据库] [MobiLink 客户端]”一节第 535 页。
priority	返回一个值，指示连接的优先级级别。请参见“ priority 选项 [数据库] ”一节第 535 页。

属性	说明
query_mem_timeout	返回 query_mem_timeout 选项的值。请参见“ query_mem_timeout 选项 [数据库] ”一节第 536 页。
QueryBypassed	返回不使用优化程序所优化的请求数。
QueryBypassedCosted	返回优化程序旁路使用开销计算所处理的请求数。
QueryBypassedHeuristic	返回优化程序旁路使用试探法所处理的请求数。
QueryBypassedOptimized	返回最初由优化程序旁路处理、随后由 SQL Anywhere 优化程序完全优化的请求数。
QueryCachedPlans	返回当前为连接高速缓存的查询执行计划数。
QueryCachePages	返回用于高速缓存执行计划的高速缓存页数。
QueryDescribedBypass	返回优化程序旁路所处理的描述请求数。
QueryDescribedOptimizer	返回优化程序所处理的描述请求数。
QueryHeapPages	返回用于查询处理（散列和排序操作）的高速缓存页数。
QueryJHToJNLOptUsed	返回散列连接转换为嵌套循环连接的次数。
QueryLowMemoryStrategy	返回服务器由于内存不足的情况而在执行期间改变其执行计划的次数。策略可能会因当前可用内存比优化程序预计的少或执行计划所需的内存比优化程序预计的多而发生变化。
QueryMemActiveCurr	返回主动使用查询内存的请求数。
QueryMemGrantFailed	返回某一请求等待但未能获得查询内存的总次数。
QueryMemGrantGranted	返回当前授予请求的页数。
QueryMemGrantRequested	返回任一请求尝试获取查询内存的总次数。
QueryMemGrantWaited	返回任一请求等待查询内存的总次数。
QueryMemGrantWaiting	返回等待查询内存的当前请求数。
QueryOpened	返回要执行的 OPEN 请求数。
QueryOptimized	返回经过完全优化的请求数。
QueryReused	返回已从计划高速缓存中重用的请求数。

属性	说明
QueryRowsBufferFetch	返回使用缓冲获取的行数。
QueryRowsMaterialized	返回查询处理期间写入工作表的行数。
quoted_identifier	如果将括在双引号中的字符串解释为标识符，则返回 On；如果将其解释为文字字符串，则返回 Off。请参见“ quoted_identifier 选项 [兼容性] ”一节第 537 页。
read_past_deleted	如果顺序扫描在隔离级别 1 和 2 跳过未提交的已删除行，则返回 On；如果顺序扫描在隔离级别 1 和 2 阻塞于未提交的已删除行，则返回 Off。请参见“ read_past_deleted 选项 [数据库] ”一节第 538 页。
recovery_time	返回数据库服务器从系统故障中恢复最长将要花费的时间（以分钟为单位）。请参见“ recovery_time 选项 [数据库] ”一节第 538 页。
RecursiveIterations	返回递归联合的迭代次数。
RecursiveIterationsHash	返回递归散列连接使用散列策略的次数。
RecursiveIterationsNested	返回递归散列连接使用嵌套循环策略的次数。
RecursiveJNLMisses	返回递归散列连接的索引探测高速缓存未命中次数。
RecursiveJNLProbes	返回递归散列连接尝试索引探测的次数。
remote_idle_timeout	返回 Web 服务客户端过程和函数允许的非活动时间（以秒为单位）。请参见“ remote_idle_timeout 选项 [数据库] ”一节第 539 页。
replicate_all	如果数据库在 Replication Server 安装中充当主站点，则返回 On；否则，返回 Off。请参见“ replicate_all 选项 [复制代理] ”一节第 539 页。
ReqCountActive	返回已处理的请求数；如果将 RequestTiming 服务器属性设置为 Off，则返回 NULL。请参见“ -zt 服务器选项 ”一节第 233 页。
ReqCountBlockContention	返回连接等待原子访问的次数；如果未指定 -zt 选项，则返回 NULL。请参见“ -zt 服务器选项 ”一节第 233 页。
ReqCountBlockIO	返回连接等待 I/O 完成的次数；如果未指定 -zt 选项，则返回 NULL。请参见“ -zt 服务器选项 ”一节第 233 页。

属性	说明
ReqCountBlockLock	返回连接等待锁的次数；如果未指定 -zt 选项，则返回 NULL。请参见“ -zt 服务器选项 ”一节第 233 页。
ReqCountUnscheduled	返回连接等待调度的次数；如果未指定 -zt 选项，则返回 NULL。请参见“ -zt 服务器选项 ”一节第 233 页。
ReqStatus	<p>返回请求状态。它可以是以下各值之一：</p> <ul style="list-style-type: none"> ● Idle 连接当前未处理请求。 ● Unscheduled* 连接有工作要做并且正在等待工作线程。 ● BlockedIO* 连接被阻塞，正在等待 I/O。 ● BlockedContention* 连接被阻塞，正在等待访问共享数据库服务器数据结构。 ● BlockedLock 连接被阻塞，正在等待锁定的对象。 ● Executing 连接正在执行请求。 <p>仅当已使用 -zt 服务器选项为数据库服务器开启了请求计时信息记录功能时，才会返回标有星号 (*) 的值。如果未记录请求计时信息（缺省设置），则会将这些值报告为 Executing。</p> <p>有关详细信息，请参见“-zt 服务器选项”一节第 233 页。</p>
ReqTimeActive	返回处理请求所用的时间长度；如果未指定 -zt 选项，则返回 NULL。请参见“ -zt 服务器选项 ”一节第 233 页。
ReqTimeBlockContention	返回等待原子访问所用的时间长度；如果将 RequestTiming 服务器属性设置为 Off，则返回 NULL。请参见“ -zt 服务器选项 ”一节第 233 页。
ReqTimeBlockIO	返回等待 I/O 完成所用的时间长度；如果未指定 -zt 选项，则返回 NULL。请参见“ -zt 服务器选项 ”一节第 233 页。
ReqTimeBlockLock	返回等待锁所用的时间长度；如果未指定 -zt 选项，则返回 NULL。请参见“ -zt 服务器选项 ”一节第 233 页。
ReqTimeUnscheduled	返回未调度的时间长度；如果未指定 -zt 选项，则返回 NULL。请参见“ -zt 服务器选项 ”一节第 233 页。
ReqType	返回上次请求的类型。
request_timeout	返回单个请求可以运行的最长时间。请参见“ request_timeout 选项 [数据库] ”一节第 540 页。

属性	说明
RequestsReceived	返回客户端/服务器通信请求数或往返次数。它与 PacketsReceived 的不同之处在于：多包请求将计为一个请求，并且不包括活动包。
return_date_time_as_string	如果将日期、时间和时间戳值作为字符串返回给应用程序，则返回 On；如果将其作为日期或时间数据类型返回，则返回 Off。请参见“ return_date_time_as_string 选项 [数据库] ”一节第 541 页。
Rlbc	已处理的回退请求数。
rollback_on_deadlock	当在 UPDATE 或 DELETE 之后执行参照完整性操作时，返回 After；如果在 UPDATE 或 DELETE 之前执行参照完整性操作，则返回 Before。请参见“ rollback_on_deadlock [数据库] ”一节第 542 页。
RollbackLogPages	返回回退日志中的页数。
row_counts	如果行计数始终准确，则返回 On；如果行计数通常为估计值，则返回 Off。请参见“ row_counts 选项 [数据库] ”一节第 542 页。
scale	返回连接的十进制和数值小数位数。请参见“ scale 选项 [数据库] ”一节第 543 页。
secure_feature_key	存储用于启用和禁用数据库服务器功能的键值。选择此属性值始终都会返回一个空字符串。
ServerNodeAddress	返回客户端/服务器连接中服务器的节点。当客户端和服务器的同一台计算机上时，将会返回一个空字符串。
ServerPort	返回数据库服务器的 TCP/IP 端口号或 0。
SessionCreateTime	返回创建 HTTP 会话的时间。
SessionID	返回连接的会话 ID（如果已定义），否则，返回一个空字符串。
SessionLastTime	返回 HTTP 会话的上次请求时间。
SessionTimeout	返回 HTTP 会话在非活动期间持续的时间（单位为分钟）。请参见“ sa_set_http_option 系统过程 ”一节《 SQL Anywhere 服务器 - SQL 参考 》。
SnapshotCount	返回与连接相关联的快照数。

属性	说明
sort_collation	如果 ORDER BY 子句保持不变，则返回 Internal；否则，返回归类名或归类 ID。请参见“ sort_collation 选项 [数据库] ”一节第 544 页。
SortMergePasses	返回在排序过程中使用的合并遍数。
SortRowsMaterialized	返回写入排序工作表的行数。
SortRunsWritten	返回在排序过程中写入的排序运行数。
SortSortedRuns	返回在运行形成过程中创建的排序运行数。
SortWorkTables	返回为排序创建的工作表数。
sql_flagger_error_level	<p>返回以下各值之一，指示将哪个不属于指定 SQL/2003 集的 SQL 标记为错误：</p> <ul style="list-style-type: none"> ● E 标记非入门级 SQL/2003 语法的语法 ● I 标记非中级 SQL/2003 语法的语法 ● F 标记非完整 SQL/2003 语法的语法 ● W 允许所有支持的语法 <p>有关详细信息，请参见“sql_flagger_error_level 选项 [兼容性]”一节第 545 页。</p>
sql_flagger_warning_level	<p>返回以下各值之一，指示将哪个不属于指定 SQL/2003 集的 SQL 标记为警告：</p> <ul style="list-style-type: none"> ● E 标记非入门级 SQL/2003 语法的语法 ● I 标记非中级 SQL/2003 语法的语法 ● F 标记非完整 SQL/2003 语法的语法 ● W 允许所有支持的语法 <p>有关详细信息，请参见“sql_flagger_warning_level 选项 [兼容性]”一节第 546 页。</p>
StatementDescribes	返回各个 DESCRIBE 请求所处理的语句总数。
StatementPostAnnotates	返回由语义查询转换阶段所处理的语句数。
StatementPostAnnotatesSimple	返回由语义查询转换阶段处理但跳过了某些语义转换的语句数。
StatementPostAnnotatesSkipped	返回已完全跳过语义查询转换阶段的语句数。

属性	说明
string_rtruncation	如果在截断字符串时产生错误，则返回 On；如果不产生错误且字符串是以静默方式截断的，则返回 Off。请参见“ string_rtruncation 选项 [兼容性] ”一节第 549 页。
subsume_row_locks	如果数据库服务器获得表的各个行锁，则返回 On；否则，返回 Off。请参见“ subsume_row_locks 选项 [数据库] ”一节第 550 页。
suppress_tds_debugging	如果 TDS 调试信息出现在数据库服务器消息窗口中，则返回 Off；如果调试信息不出现在数据库服务器消息窗口中，则返回 On。请参见“ suppress_tds_debugging 选项 [数据库] ”一节第 551 页。
synchronize_mirror_on_commit	如果在提交时同步数据库镜像服务器，则返回 On；否则，返回 Off。请参见“ synchronize_mirror_on_commit 选项 [数据库] ”一节第 551 页。
tds_empty_string_is_null	如果对于 TDS 连接将空字符串作为 NULL 返回，则返回 On；如果对于 TDS 连接返回包含单个空字符串的字符串，则返回 Off。请参见“ tds_empty_string_is_null 选项 [数据库] ”一节第 552 页。
temp_space_limit_check	如果数据库服务器检查连接可用的临时空间量，则返回 On；如果数据库服务器不检查连接可用的空间量，则返回 Off。请参见“ temp_space_limit_check 选项 [数据库] ”一节第 552 页。
TempTablePages	返回用于临时表的临时文件中的页数。
time_format	返回从数据库中检索的时间所使用的字符串格式。请参见“ time_format 选项 [兼容性] ”一节第 553 页。
time_zone_adjustment	返回为显示连接本地时间而必须对协调通用时间（Coordinated Universal Time，简称 UTC）增加的分钟数。请参见“ time_zone_adjustment 选项 [数据库] ”一节第 554 页。
timestamp_format	返回为显示连接本地时间而必须对协调通用时间（Coordinated Universal Time，简称 UTC）增加的分钟数。请参见“ timestamp_format 选项 [兼容性] ”一节第 554 页。
TimeZoneAdjustment	返回为显示连接本地时间而必须对协调通用时间（Coordinated Universal Time，简称 UTC）增加的分钟数。请参见“ time_zone_adjustment 选项 [数据库] ”一节第 554 页。

属性	说明
TransactionStartTime	返回一个字符串，其中包含在 COMMIT 或 ROLLBACK 之后首次修改数据库的时间；如果自上次执行 COMMIT 或 ROLLBACK 以来未对数据库进行过任何修改，则返回一个空字符串。
truncate_timestamp_values	如果时间戳值中使用的小数位受限制，则返回 On；否则，返回 Off。请参见“truncate_timestamp_values 选项 [数据库] [MobiLink 客户端]”一节第 556 页。
tsql_outer_joins	如果可以在 DML 语句中使用 Transact-SQL 外连接，则返回 On。请参见“tsql_outer_joins 选项 [兼容性]”一节第 557 页。
tsql_variables	如果可以使用 @ 符号代替冒号作为嵌入式 SQL 中主机变量名的前缀，则返回 On；否则，返回 Off。请参见“tsql_variables 选项 [兼容性]”一节第 557 页。
UncommitOp	返回未提交的操作数。
updatable_statement_isolation	返回当 isolation_level 选项设置为 readonly-statement-snapshot（只读语句快照）时，可更新语句使用的隔离级别（0、1、2 或 3）。请参见“updatable_statement_isolation 选项 [数据库]”一节第 557 页。
update_statistics	此属性保留供系统使用。请不要更改此选项的设置。
upgrade_database_capability	此属性保留供系统使用。请不要更改此选项的设置。
user_estimates	<p>返回以下各值之一，这些值控制查询优化程序是考虑还是忽略查询谓语句中的选择性估计：</p> <ul style="list-style-type: none"> ● Enabled 考虑所有由用户提供的选择性估计。也可以使用 On 开启此选项。 ● Override-Magic 只有优化程序要不然会选择采取最后的手段，即试探值（也称魔术值），才会考虑并使用用户选择性估计。 ● 已禁用 忽略所有用户估计，并在没有其它可用的估计数据时使用魔术值。也可以使用 Off 关闭此选项。 <p>有关详细信息，请参见“user_estimates 选项 [数据库]”一节第 559 页。</p>

属性	说明
UserAppInfo	返回连接字符串中的 AppInfo 连接参数所指定的字符串。 有关详细信息，请参见“ AppInfo 连接参数 [APP] ”一节第 249 页。
UserID	返回连接的用户 ID。
UtilCmdsPermitted	返回 On 或 Off，指示连接是否允许使用实用程序命令（如 CREATE DATABASE、DROP DATABASE 和 RESTORE DATABASE）。请参见“ -gu 服务器选项 ”一节第 188 页。
verify_password_function	返回用于口令验证的函数名（如果已指定）。请参见“ verify_password_function 选项 [数据库] ”一节第 560 页。
wait_for_commit	如果数据库要在下一条 COMMIT 语句之后才会检查外键完整性，则返回 On。否则，返回 Off，并且在插入、更新或删除所有不是使用 check_on_commit 选项创建的外键时都会对其进行检查。请参见“ wait_for_commit 选项 [数据库] ”一节第 564 页。
WaitStartTime	返回连接开始等待的时间（如果连接没有处于等待状态则返回一个空字符串）。
WaitType	返回等待的原因（如果可用）。WaitType 的可能值有： <ul style="list-style-type: none"> ● lock 如果连接在等待锁定则返回此值。 ● waitfor 如果连接在执行 waitfor 语句则返回此值。 ● 空字符串 如果连接没有处于等待状态，或等待的原因没有提供，则返回此值。
webservice_namespace_host	返回要在生成的 WSDL 文档内用作 XML 命名空间的主机名（如果已指定）。请参见“ webservice_namespace_host 选项 [数据库] ”一节第 564 页。

数据库服务器属性

下表列出了在整个数据库服务器上应用的属性。

可以使用 `PROPERTY` 系统函数检索单个属性的值，也可使用 `sa_eng_properties` 系统过程检索所有数据库服务器属性的值。属性名称不区分大小写。

示例

◆ 检索数据库服务器属性的值

- 使用 `PROPERTY` 系统函数。例如，以下语句返回用于全局服务器数据结构的高速缓存页数：

```
SELECT PROPERTY ( 'MainHeapPages' );
```

◆ 检索所有服务器属性的值：

- 使用 `sa_eng_properties` 系统过程：

```
CALL sa_eng_properties;
```

另请参见

- “`PROPERTY` 函数 [System]” 一节 《SQL Anywhere 服务器 - SQL 参考》
- “连接属性” 一节第 566 页
- “数据库属性” 一节第 601 页

说明

属性	说明
ActiveReq	返回当前处理请求的服务器线程数。
AvailIO	返回当前可用的 I/O 控制块数。
BuildChange	保留。
BuildClient	保留。
BuildProduction	如果数据库服务器是为了用于生产而编译的，则返回 Yes；如果数据库服务器为调试内部版本，则返回 No。
BuildReproducible	保留。
BytesReceived	返回在客户端/服务器通信期间所收到的字节数。此值对于 HTTP 和 HTTPS 连接会进行更新。
BytesReceivedUncomp	返回在禁用压缩的情况下，在客户端/服务器通信期间所收到的字节数。（在禁用压缩的情况下该值与 BytesReceived 的值相同。）

属性	说明
BytesSent	返回在客户端/服务器通信期间所发送的字节数。此值对于 HTTP 和 HTTPS 连接会进行更新。
BytesSentUncomp	返回在禁用压缩的情况下，在客户端/服务器通信期间所发送的字节数。（在禁用压缩的情况下该值与 BytesSent 的值相同。）
CacheAllocated	返回已为服务器数据结构分配的高速缓存页数。
CacheFile	返回用于存放数据库文件中数据的高速缓存页数。
CacheFileDirty	返回处于脏状态的高速缓存页数（需要写操作）。
CacheFree	返回未在使用的高速缓存页数。
CacheHits	返回数据库页查找次数。
CachePanics	返回高速缓存管理器未能找到要分配页的次数。
CachePinned	返回固定高速缓存页数。
CacheRead	返回高速缓存读取次数。
CacheReplacements	返回高速缓存中已替换的页数。
CacheScavenges	返回高速缓存管理器清除回收以分配页的次数。
CacheScavengeVisited	返回在清除回收以分配页期间访问的页数。
CacheSizingStatistics	如果服务器在重新调整高速缓存大小时显示高速缓存大小调整统计信息，则返回 Yes；否则，返回 No。请参见“ -cs 服务器选项 ”一节第 165 页。
CarverHeapPages	返回供短期使用（如查询优化）的堆页数。
CharSet	返回数据库服务器所使用的 CHAR 字符集。
ClientStmtCacheHits	返回因客户端语句高速缓存而不需要进行的准备的数目。如果禁用了客户端语句高速缓存，则为必需的附加准备数。请参见“ max_client_statements_cached 选项 [数据库] ”一节第 516 页。
ClientStmtCacheMisses	返回客户端语句高速缓存中再次准备的语句数。即为由于模式更改、数据库选项设置或 DROP VARIABLE 语句等原因而使本打算重用的某个高速缓存的语句无法重用的次数。请参见“ max_client_statements_cached 选项 [数据库] ”一节第 516 页。
CollectStatistics	返回 Yes 或 No，指示数据库服务器是否正在收集性能统计信息。请参见“ -k 服务器选项 ”一节第 190 页。

属性	说明
CommandLine	返回用于启动数据库服务器的命令行。 如果数据库的加密密钥是使用 -ek 选项指定的，则该密钥在由此属性返回的值中将以星号字符串常量代替。
CompactPlatformVer	返回 PlatformVer 属性的精简版本。
CompanyName	返回拥有此软件的公司的名称。
ConnsDisabled	返回 Yes 或 No，指示用于禁用新连接的服务器选项的当前设置。请参见“ sa_server_option 系统过程 ”一节《 SQL Anywhere 服务器 - SQL 参考 》。
ConsoleLogFile	如果指定了 -o 选项，则返回用于记录数据库服务器消息的文件的名称；否则，返回一个空字符串。请参见“ -o 服务器选项 ”一节第 196 页和“ 记录数据库服务器操作 ”一节第 40 页。
ConsoleLogMaxSize	返回用于记录数据库服务器消息的文件的最大大小（以字节为单位）。请参见“ -os 服务器选项 ”一节第 198 页。
CurrentCacheSize	返回当前高速缓存大小（以千字节为单位）。
DebuggingInformation	如果服务器显示用于进行排除故障的诊断消息，则返回 Yes；否则，返回 No。请参见“ -z 服务器选项 ”一节第 226 页。
DefaultCollation	返回如果没有明确指定，将用于新数据库的归类。
DefaultNcharCollation	返回服务器计算机上缺省 NCHAR 归类的名称（如果安装了 ICU，则为 UCA；否则为 UTF8BIN）。
DiskRead	返回磁盘读取次数。
DiskReadHintScatterLimit	返回对分散读取提示大小（以字节为单位）的强加限制。
DiskRetryRead	返回磁盘读取重试的次数。
DiskRetryReadScatter	返回分散读取的磁盘读取重试次数。
DiskRetryWrite	返回磁盘写入重试的次数。
EventTypeDesc	返回与给定事件类型 ID 相关联的系统事件类型说明。
EventTypeName	返回与给定事件类型 ID 相关联的系统事件类型名称。
ExchangeTasks	返回当前用于并行执行查询的任务数。

属性	说明
ExchangeTasksCompleted	返回自数据库服务器启动以来已用于查询内并行机制的内部任务总数。请参见“ 查询执行期间的并行 ”一节《 SQL Anywhere 服务器 - SQL 的用法 》。
FipsMode	如果在启动数据库服务器时指定了 <code>-fips</code> 选项，则返回 Yes；否则，返回 No。
FirstOption	返回一个数字，表示与某个数据库选项相对应的第一个连接属性。
FreeBuffers	返回可用的网络缓冲区数。
FunctionMaxParms	<p>返回最多可为函数指定的参数个数。函数通过 <i>function-number</i> 所指定的值进行标识，该值是一个正整数。例如：</p> <pre>SELECT PROPERTY ('FunctionMaxParms', function-number);</pre> <p>请注意，<i>function-number</i> 在不同的版本之间可能会发生变化。</p>
FunctionMinParms	<p>返回必须为函数指定的最少参数个数。函数通过 <i>function-number</i> 所指定的值进行标识，该值是一个正整数。例如：</p> <pre>SELECT PROPERTY ('FunctionMaxParms', function-number);</pre> <p>请注意，<i>function-number</i> 在不同的版本之间可能会发生变化。</p>
FunctionName	<p>返回通过 <i>function-number</i> 指定的值（该值是一个正整数）所标识的函数的名称：</p> <pre>SELECT PROPERTY ('FunctionName', function-number);</pre> <p>请注意，<i>function-number</i> 在不同的版本之间可能会发生变化。</p>
HeapsCarver	返回供短期使用（如查询优化）的堆数。
HeapsLocked	返回高速缓存中当前锁定的可重定位堆数。
HeapsQuery	返回用于查询处理（散列和排序操作）的堆数。
HeapsRelocatable	返回可重定位堆数。
HttpAddresses	<p>返回服务器为 HTTP 连接正在监听的 TCP/IP 地址的列表（以分号分隔）。例如：</p> <pre>(::1):80;127.0.0.1:80</pre>

属性	说明
HttpNumActiveReq	返回正在主动处理 HTTP 请求的 HTTP 连接数。不包括已发送其响应的 HTTP 连接。
HttpNumConnections	返回数据库服务器中当前打开的 HTTP 连接数。这些连接可能正在主动处理请求，或者正在长寿命（保持活动状态）连接队列中等待。
HttpNumSessions	返回数据库服务器中活动的和休眠的 HTTP 会话数量。
HttpPorts	返回 Web 服务器的 HTTP 端口号列表（以逗号分隔）。
HttpsAddresses	返回服务器为 HTTPS 连接正在监听的 TCP/IP 地址的列表（以分号分隔）。例如： <code>(::1):443;127.0.0.1:443</code>
HttpsNumActiveReq	返回正在主动处理 HTTPS 请求的安全 HTTPS 连接数量。不包括已发送其响应的 HTTPS 连接。
HttpsNumConnections	返回数据库服务器中当前打开的 HTTPS 连接数。这些连接可能正在主动处理请求，或者正在长寿命（保持活动状态）连接队列中等待。
HttpsPorts	返回 Web 服务器的 HTTPS 端口号列表（以逗号分隔）。
IdleTimeout	返回缺省空闲超时值。请参见“ -ti 服务器选项 ”一节第 212 页。
IsEccAvailable	如果安装了 ECC DLL，则返回 Yes；否则，返回 No。
IsFipsAvailable	如果安装了 FIPS DLL，则返回 Yes；否则，返回 No。
IsNetworkServer	如果连接到网络数据库服务器，则返回 Yes；如果连接到个人数据库服务器，则返回 No。
IsRsaAvailable	如果安装了 RSA DLL，则返回 Yes；否则，返回 No。
IsRuntimeServer	如果连接到受限的桌面运行时数据库服务器，则返回 Yes；否则，返回 No。
IsService	如果数据库服务器是作为服务来运行，则返回 Yes，否则返回 No。
Language	返回服务器的地区语言。
LastConnectionProperty	返回表示最后一个连接属性的数字。
LastDatabaseProperty	返回表示最后一个数据库属性的数字。

属性	说明
LastOption	返回一个数字，表示与某个数据库选项相对应的最后一个连接属性。
LastServerProperty	返回表示最后一个服务器属性的数字。
LegalCopyright	返回软件的版权字符串。
LegalTrademarks	返回软件的商标信息。
LicenseCount	返回被授权的存取设备或处理器数量。
LicensedCompany	返回被授权公司的名称。
LicensedUser	返回被授权用户的名称。
LicenseType	返回许可类型。可以是联网存取设备（每存取设备）或基于 CPU 的类型。
LivenessTimeout	返回客户端活动超时缺省值。请参见“ -tl 服务器选项 ”一节第 212 页。
LockedCursorPages	返回用于使游标堆在内存中保持固定的页数。
LockedHeapPages	返回高速缓存中锁定的堆页数。
MachineName	返回运行数据库服务器的计算机名称。通常，这是计算机的主机名。
MainHeapBytes	返回用于全局服务器数据结构的字节数。
MainHeapPages	返回用于全局服务器数据结构的页数。
MapPhysicalMemoryEng	返回使用 Address Windowing Extensions 映射到高速缓存中物理内存的数据库页地址空间窗口的数量。
MaxCacheSize	返回所允许的最大高速缓存大小（以千字节为单位）。
MaxConnections	返回服务器允许的最大并发连接数。对于个人服务器，此值在缺省情况下为 10。对于网络服务器，此值在缺省情况下约为 32000。可使用 -gm 服务器选项 降低此值。请参见“ -gm 服务器选项 ”一节第 182 页。 受计算机资源所限，网络服务器的连接数通常低于缺省值。
MaxEventType	返回最大有效事件类型 ID。

属性	说明
MaxMessage	不建议使用。返回可从数据库服务器消息窗口检索的当前最大行号。它表示数据库服务器消息窗口中显示的最新消息。
MaxRemoteCapability	返回最大有效功能 ID。
Message, <i>linenumber</i>	<p>不建议使用。返回数据库服务器消息窗口中的某一行，前面带有消息出现的日期和时间。第二个参数指定行号。</p> <p>由 <code>PROPERTY("message")</code> 返回的值是写入数据库服务器消息窗口的第一行输出。调用 <code>PROPERTY("message", n)</code> 会返回服务器输出的第 <code>n</code> 行（0 为第一行）。缓冲区是有限的，因此随着消息不断被生成，前面的行可能被删除且不再存储在内存中。在这种情况下，返回 NULL。</p>
MessageCategoryLimit	返回可使用 <code>sa_server_messages</code> 系统过程检索的各个严重级和类别的最小消息数。缺省值为 400。请参见“ sa_server_messages 系统过程 ”一节《 SQL Anywhere 服务器 - SQL 参考 》。
MessageText, <i>linenumber</i>	不建议使用。返回与数据库服务器消息窗口中所指定的行号相关联的文本，前面不带日期和时间。第二个参数指定行号。
MessageTime, <i>linenumber</i>	不建议使用。返回与数据库服务器消息窗口中所指定的行号相关联的日期和时间。第二个参数指定行号。
MessageWindowSize	不建议使用。返回可从数据库服务器消息窗口检索的最大行数。
MinCacheSize	返回所允许的最小高速缓存大小（以千字节为单位）。
MultiPacketsReceived	返回在客户端/服务器通信期间所收到的多包请求数。
MultiPacketsSent	返回在客户端/服务器通信期间所发送的多包请求数。
MultiPageAllocs	返回多页高速缓存分配数。
MultiProgrammingLevel	返回服务器最多可以处理的并发任务数。如果并发任务数大于此值，则会将请求送入队列。可使用 <code>-gn</code> 服务器选项对此进行更改。请参见“ -gn 服务器选项 ”一节第 182 页。
Name	如果指定了用于连接到数据库的服务器的替代名称，则返回该名称；否则，返回实际服务器名。请参见“ -sn 数据库选项 ”一节第 242 页。

属性	说明
NativeProcessorArchitecture	<p>返回一个字符串，它标识了可对处理器进行模拟的平台（如 Win64 上的 X86）上的本地处理器类型。在所有其它情况下，其返回值与 property('ProcessorArchitecture') 的返回值相同。</p> <p>值可以包括：</p> <ul style="list-style-type: none"> ● 32 位 Windows（Windows Mobile 除外） - X86 ● Windows Mobile - ARM ● 64 位 Windows - X86_64 ● Solaris - SPARC 或 X86_64 ● AIX - PPC ● MAC OS - X86 或 X86_64 ● HP - IA64 ● Linux - X86 或 X86_64 <p>有关支持的平台的完整列表，请参见 http://www.sybase.com/detail?id=1062617。</p>
NumLogicalProcessors	返回服务器计算机上已启用的逻辑处理器数（包括内核和超线程）。
NumLogicalProcessorsUsed	返回数据库服务器将要使用的逻辑处理器数。在 Windows 上，可使用 -gtc 选项更改所使用的逻辑处理器数。请参见“-gtc 服务器选项”一节第 186 页。
NumPhysicalProcessors	返回服务器计算机上已启用的物理处理器数。此值是用 NumLogicalProcessors 除以每个物理处理器的内核或超线程数得到的。在某些非 Windows 平台上，可能会将内核或超线程计为物理处理器。
NumPhysicalProcessorsUsed	返回数据库服务器将要使用的物理处理器数。在某些平台上，个人服务器仅限使用一个处理器。在 Windows 上，可使用 -gt 选项更改网络数据库服务器所使用的物理处理器数。请参见“-gt 服务器选项”一节第 185 页。
OmniIdentifier	此属性保留供系统使用。请不要更改此选项的设置。
PacketsReceived	返回所收到的客户端/服务器通信包数。此值对于 HTTP 和 HTTPS 连接不会进行更新。
PacketsReceivedUncomp	返回在禁用压缩的情况下，在客户端/服务器通信期间所收到的包数。（在禁用压缩的情况下，该值与 PacketsReceived 的值相同。）
PacketsSent	返回所发送的客户端/服务器通信包数。此值对于 HTTP 和 HTTPS 连接不会进行更新。

属性	说明
PacketsSentUncomp	返回在禁用压缩的情况下，在客户端/服务器通信期间所发送的包数。（在禁用压缩的情况下，该值与 PacketsSent 的值相同。）
PageSize	返回数据库服务器高速缓存页的大小。可以使用 -gp 选项设置该值，否则为命令行中指定数据库的数据库页面的最大值。
PeakCacheSize	返回高速缓存在当前会话中已达到的最大值（以千字节为单位）。
Platform	返回运行软件的操作系统。例如，如果您正运行在 Windows 2000 上，则此属性返回 Windows2000。
PlatformVer	返回运行软件的操作系统，其中包括内部版本号、服务包等。例如，可以返回 Windows 2000 Build 2195 Service Pack 3。
ProcessCPU	<p>返回数据库服务器进程的 CPU 使用情况。这些值以秒为单位。在 Windows 和 Unix 上支持此属性。在 Windows Mobile 上不支持此属性。</p> <p>此属性的返回值是自启动数据库服务器以来的累计值。该值与应用程序（如 Windows 任务管理器或 Windows 性能监控器）所返回的瞬时值不匹配。</p>
ProcessCPUSystem	<p>返回数据库服务器进程 CPU 的系统 CPU 使用情况。这是数据库服务器花费在操作系统内核中的 CPU 时间。这些值以秒为单位。在 Windows 和 Unix 上支持此属性。在 Windows Mobile 上不支持此属性。</p> <p>此属性的返回值是自启动数据库服务器以来的累计值。该值与应用程序（如 Windows 任务管理器或性能监控器）所返回的瞬时值不匹配。</p>
ProcessCPUUser	<p>返回数据库服务器进程的用户 CPU 使用情况。这些值以秒为单位。这包括数据库服务器花费在操作系统内核中的 CPU 时间。在 Windows 和 Unix 上支持此属性。在 Windows Mobile 上不支持此属性。</p> <p>此属性的返回值是自启动数据库服务器以来的累计值。该值与应用程序（如 Windows 任务管理器或性能监控器）所返回的瞬时值不匹配。</p>

属性	说明
ProcessorArchitecture	返回用于标识处理器类型的字符串。值包括： <ul style="list-style-type: none"> ● 32 位 Windows (Windows Mobile 除外) - X86 ● 64 位 Windows - X86_64 ● Windows Mobile - ARM ● Solaris - SPARC 或 X86_64 ● AIX - PPC ● MAC OS - X86 ● HP - IA64 ● Linux - X86 或 X86_64
ProductName	返回软件的名称。
ProductVersion	返回正在运行的软件版本。
ProfileFilterConn	如果开启对某特定连接的过程分析，则返回被监控连接的 ID。否则，返回一个空字符串。可使用 sa_server_option 过程来控制用户所进行的过程分析。请参见“sa_server_option 系统过程”一节《SQL Anywhere 服务器 - SQL 参考》。
ProfileFilterUser	如果开启对某特定用户的过程分析，则返回被监控用户的名称。否则，返回一个空字符串。可使用 sa_server_option 过程来控制用户所进行的过程分析。请参见“sa_server_option 系统过程”一节《SQL Anywhere 服务器 - SQL 参考》。
QueryHeapPages	返回用于查询处理（散列和排序操作）的高速缓存页数。
QueryMemActiveCurr	返回主动使用查询内存的请求数。
QueryMemActiveEst	返回在数据库服务器处于稳定状态时对主动使用查询内存的请求数平均值的估计。
QueryMemActiveMax	返回主动允许使用查询内存的最大请求数。
QueryMemExtraAvail	返回可用于在基础内存密集型授权之外进行授权的内存量。
QueryMemGrantBase	返回授权给所有请求的最小内存量。
QueryMemGrantBaseMI	返回授权给内存密集型请求的最小内存量。
QueryMemGrantExtra	返回除 QueryMemGrantBaseMI 以外可分发到活动内存密集型请求的查询内存页数。
QueryMemGrantFailed	返回某一请求等待但未能获得查询内存的总次数。
QueryMemGrantGranted	返回当前授予请求的页数。

属性	说明
QueryMemGrantRequested	返回任一请求尝试获取查询内存的总次数。
QueryMemGrantWaited	返回任一请求等待查询内存的总次数。
QueryMemGrantWaiting	返回等待查询内存的当前请求数。
QueryMemPages	返回可用于查询执行算法的内存量，以页数表示。
QueryMemPercentOfCache	返回可用于查询执行算法的内存量，以最大高速缓存大小的百分比表示。
QuittingTime	返回服务器的关机时间。如果没有指定，则该值为 none。
RememberLastPlan	如果服务器正在记录优化程序返回的最后一个查询优化计划，则返回 Yes。请参见“ -zp 服务器选项 ”一节第 230 页。
RememberLastStatement	如果服务器正在记录每个连接所准备的最后一条语句，则返回 Yes；否则，返回 No。请参见“ -zl 服务器选项 ”一节第 227 页。
RemoteCapability	返回与给定功能 ID 相关联的远程功能的名称。
RemoteputWait	返回远程放置等待数。
Req	返回为使服务器可以处理新请求或继续处理现有请求而进入该服务器的次数。
RequestFilterConn	返回正在为其过滤记录信息的连接的 ID，否则，返回 -1。
RequestFilterDB	返回正在为其过滤记录信息的数据库的 ID，否则，返回 -1。
RequestLogFile	返回请求记录文件名。如果没有任何请求记录，则会返回一个空字符串。请参见“ sa_server_option 系统过程 ”一节《 SQL Anywhere 服务器 - SQL 参考 》。
RequestLogging	返回 SQL、PLAN、HOSTVARS、PROCEDURES、TRIGGERS、OTHER、BLOCKS、REPLACE、ALL 或 NONE 之一，指示请求记录的当前设置。请参见“ sa_server_option 系统过程 ”一节《 SQL Anywhere 服务器 - SQL 参考 》。
RequestLogMaxSize	返回请求日志文件的最大大小。请参见“ -zs 服务器选项 ”一节第 232 页。
RequestLogNumFiles	返回保留的请求日志文件数。请参见“ sa_server_option 系统过程 ”一节《 SQL Anywhere 服务器 - SQL 参考 》。

属性	说明
RequestsReceived	返回客户端/服务器通信请求数或往返次数。它与 PacketsReceived 的不同之处在于：多包请求将计为一个请求，并且不包括活动包。
RequestTiming	如果开启请求计时，则返回 Yes；否则，返回 No。可使用 -zt 数据库服务器选项开启请求计时。请参见“ -zt 服务器选项 ”一节第 233 页。
SendFail	返回底层通信协议发送数据包失败的次数。
ServerEdition	<p>返回数据库服务器许可的类型。值包括：</p> <ul style="list-style-type: none"> ● Education ● Full Developer Evaluation ● Web Authenticated ● RunTime ● IQ <p>如果您对于以下任一功能具有单独的许可，则相应的字符串会被添加到返回的许可字符串。</p> <ul style="list-style-type: none"> ● HighAvailability 请参见“SQL Anywhere 高可用性组件”一节《SQL Anywhere 11 - 简介》。 ● InMemory 请参见“SQL Anywhere 内存模式选项”一节《SQL Anywhere 11 - 简介》。 ● ECC 请参见“SQL Anywhere 安全性组件”一节《SQL Anywhere 11 - 简介》。 ● FIPS 请参见“SQL Anywhere 安全性组件”一节《SQL Anywhere 11 - 简介》。
ServerName	返回当前连接的服务器名称。可使用此值确定哪个工作服务器当前在担当数据库镜像配置中的主服务器。请参见“ 数据库镜像简介 ”一节第 880 页。
StartDBPermission	返回 -gd 服务器选项的设置，该值可以为 DBA、all 或 none 之一。请参见“ -gd 服务器选项 ”一节第 179 页。
StartTime	返回服务器启动的日期/时间。
StreamsUsed	返回正在使用的数据库服务器流的数量。
TcpIpAddresses	<p>返回服务器为命令序列和 TDS 连接正在监听的 TCP/IP 地址的列表（以分号分隔）。例如：</p> <pre>(::1):2638;127.0.0.1:2638</pre>

属性	说明
TempDir	返回服务器存储临时文件的目录。
TimeZoneAdjustment	返回为显示服务器本地时间而必须对协调通用时间 (Coordinated Universal Time, 简称 UTC) 增加的分钟数。
TotalBuffers	返回网络缓冲区的总数。
UniqueClientAddresses	返回连接到网络服务器的唯一客户端网络地址数。
UnschReq	返回当前排队等候可用服务器线程的请求数。
WebClientLogFile	返回 Web 服务客户端日志文件的名称。请参见“ -zoc 服务器选项 ”一节第 229 页。
WebClientLogging	返回一个值，指示是否正在将 Web 服务客户端信息记录到文件中。请参见“ -zoc 服务器选项 ”一节第 229 页。

数据库属性

下表列出了数据库服务器上每个数据库的可用属性。

可以使用 DB_PROPERTY 系统函数检索单个属性的值，也可使用 sa_db_properties 系统过程检索所有数据库属性的值。属性名称不区分大小写。

示例

◆ 检索数据库属性的值：

- 使用 DB_PROPERTY 系统函数。例如，以下语句返回当前数据库的页大小：

```
SELECT DB_PROPERTY ( 'PageSize' );
```

◆ 检索所有数据库属性的值：

- 使用 sa_db_properties 系统过程：

```
CALL sa_db_properties;
```

另请参见

- “DB_PROPERTY 函数 [System]” 一节 《SQL Anywhere 服务器 - SQL 参考》
- “DB_EXTENDED_PROPERTY 函数 [System]” 一节 《SQL Anywhere 服务器 - SQL 参考》
- “数据库服务器属性” 一节第 588 页
- “连接属性” 一节第 566 页

说明

属性	说明
AccentSensitive	返回区分重音功能的状态。如果数据库区分重音，则返回 Yes；如果不区分，则返回 No；如果使用的是区分法文规则，则返回 FRENCH。
Alias	返回数据库名。
AlternateMirrorServerName	返回与数据库相关联的替代镜像服务器的名称（如果已指定）。请参见“-sm 数据库选项”一节第 241 页。
AlternateServerName	返回与数据库相关联的替代服务器名称（如果已指定）。请参见“-sn 数据库选项”一节第 242 页。

属性	说明
ArbiterState	<p>返回以下各值之一：</p> <ul style="list-style-type: none"> ● NULL 连接到某个没有镜像的数据库。 ● connected 仲裁服务器连接到主服务器。 ● disconnected 仲裁服务器未连接到主服务器。 <p>请参见“数据库镜像简介”一节第 880 页。</p>
AuditingTypes	返回当前启用的审计类型。请参见“ auditing 选项 [数据库] ”一节第 484 页
Authenticated	如果数据库已经过验证，则返回 Yes，如果数据库还未经过验证，则返回 No。
BlankPadding	如果数据库启用了空格填充，则返回 On。否则，返回 Off。
CacheHits	返回通过在高速缓存中找到相应的页而得到满足的数据库页查找次数。
CacheRead	已在高速缓存中查找的数据库页数。
CacheReadIndInt	返回已从高速缓存中读取的索引内部节点页数。
CacheReadIndLeaf	返回已从高速缓存中读取的索引叶页数。
CacheReadTable	返回已从高速缓存中读取的表页数。
CacheReadWorkTable	返回高速缓存工作表读取次数。
Capabilities	返回为数据库启用的容量位。该属性主要供技术支持使用。
CaseSensitive	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 On。否则，返回 Off。在区分大小写的数据库中，数据比较也区分大小写。此设置不影响标识符的大小写区分。口令始终区分大小写。请参见“ 区分大小写 ”一节《 SQL Anywhere 服务器 - SQL 的用法 》。
CatalogCollation	返回用于目录的归类标识符。此属性具有一些在查询属性值可以指定的扩展项。请参见“ DB_EXTENDED_PROPERTY 函数 [System] ”一节《 SQL Anywhere 服务器 - SQL 参考 》。
CharSet	<p>返回数据库的 CHAR 字符集。</p> <p>此属性具有一些在查询属性值时可以指定的扩展项。请参见“DB_EXTENDED_PROPERTY 函数 [System]”一节《SQL Anywhere 服务器 - SQL 参考》。</p>

属性	说明
CheckpointLogBitmapPagesWritten	返回检查点日志位图的写入次数。
CheckpointLogBitmapSize	返回检查点日志位图大小。
CheckpointLogCommitToDisk	返回向磁盘提交检查点日志的次数。
CheckpointLogPagesInUse	返回正在使用的检查点日志页数。
CheckpointLogPagesRelocated	返回重新定位的检查点日志页数。
CheckpointLogPagesWritten	返回已写入的检查点日志页数。
CheckpointLogSavePreimage	返回向检查点日志添加的数据库页前映像的数量。
CheckpointLogSize	返回检查点日志的大小（以页为单位）。
CheckpointLogWrites	返回检查点日志的写入次数。
CheckpointUrgency	返回数据库检查点时间设置与自上个检查点以来所经过的时间的百分比。
Checksum	如果为数据库启用了数据库页校验和，则返回 On。否则，返回 Off。对于至关重要的页始终都有校验和。
Chkpt	返回已执行的检查点数。
ChkptFlush	返回在检查点期间写出的相邻页范围数。
ChkptPage	返回事务日志检查点的数量。
CleanablePagesAdded	返回自数据库服务器启动以来已标记为待清理的页数。
CleanablePagesCleaned	返回自数据库服务器启动以来已清理的数据库页数。
CleanableRowsAdded	返回自数据库服务器启动以来已标记为待删除的行数。
CleanableRowsCleaned	返回自数据库服务器启动以来已删除的影子表行数。
Collation	返回数据库所使用的归类。有关可用归类的列表，请参见“支持的归类和替代归类”一节第 404 页。 此属性具有一些在查询属性值时可以指定的扩展项。请参见“DB_EXTENDED_PROPERTY 函数 [System]”一节《SQL Anywhere 服务器 - SQL 参考》。
CommitFile	返回服务器强制刷新磁盘高速缓存的次数。在 Windows 上，如果使用非缓冲（直接）I/O，则不需要刷新磁盘高速缓存。

属性	说明
ConnCount	返回数据库的连接数。属性值不包括用于触发事件或其它内部操作的连接，但却包含用于外部环境支持的连接。如果您要获取使用中授权连接数目的准确计数，则可执行以下语句： <code>SELECT COUNT(*) FROM sa_conn_info()</code>
ConnsDisabled	如果禁用与当前数据库的连接，则返回 On；否则，返回 Off。
CurrentRedoPos	返回要记录下一个数据库操作的事务日志文件中的当前偏移量。
CurrIO	返回服务器已发出但尚未完成的当前文件 I/O 数。
CurrRead	返回服务器已发出但尚未完成的文件读取数。
CurrWrite	返回服务器已发出但尚未完成的文件写入数。
DatabaseCleaner	返回 On 或 Off 以指示是否启用数据库清理程序。
DBFileFragments	返回数据库文件碎片数。在 Windows 上支持此属性。
DiskRead	返回已从磁盘读取的页数。
DiskReadHint	返回磁盘读取提示数。
DiskReadHintPages	返回磁盘读取提示页数。
DiskReadIndInt	返回已从磁盘读取的索引内部节点页数。
DiskReadIndLeaf	返回已从磁盘读取的索引叶页数。
DiskReadTable	返回已从磁盘读取的表页数。
DiskReadWorkTable	返回磁盘工作表读取数。
DiskRetryReadScatter	返回分散读取的磁盘读取重试次数。
DiskSyncRead	返回同步发出的磁盘读取数。
DiskSyncWrite	返回同步发出的写入数。
DiskWaitRead	返回数据库服务器等待异步读取的次数。
DiskWaitWrite	返回数据库服务器等待异步写入的次数。
DiskWrite	返回已写入磁盘的修改页数。
DiskWriteHint	返回磁盘写入提示数。

属性	说明
DiskWriteHintPages	返回磁盘收集写入提示的次数。
DriveType	<p>返回数据库文件所在驱动器的类型。该值为以下各值之一：CD、FIXED、RAMDISK、REMOTE、REMOVABLE 或 UNKNOWN。</p> <p>在 Unix 上，视 Unix 版本和驱动器类型的不同，可能无法确定驱动器类型。在这些情况下，将返回 UNKNOWN。</p> <p>此属性具有一些在查询属性值时可以指定的扩展项。请参见“DB_EXTENDED_PROPERTY 函数 [System]”一节《SQL Anywhere 服务器 - SQL 参考》。</p>
Encryption	返回用于数据库或表加密的加密类型，可为 None、Simple、AES、AES256、AES_FIPS 或 AES256_FIPS 之一。
EncryptionScope	<p>返回可以加密的数据库部分（如果有）。该值为以下各值之一：TABLE、DATABASE 或 NONE。</p> <p>TABLE 指示启用了表加密。DATABASE 指示对整个数据库进行加密。NONE 指示不启用表加密，也不对数据库进行加密。</p>
ExprCacheAbandons	返回表达式高速缓存因命中率太低而被彻底放弃的次数。
ExprCacheDropsToReadOnly	返回表达式高速缓存因命中率低而降为只读状态的次数。
ExprCacheEvicts	返回从表达式高速缓存中的退出次数。
ExprCacheHits	返回表达式高速缓存中的命中次数。
ExprCacheInserts	返回在表达式高速缓存中插入的值的个数。
ExprCacheLookups	返回在表达式高速缓存中执行的查找次数。
ExprCacheResumesOfReadWrite	返回表达式高速缓存因命中率上升而恢复读写状态的次数。
ExprCacheStarts	返回表达式高速缓存的启动次数。
ExtendDB	返回数据库文件扩展的页数。
ExtendTempWrite	返回临时文件扩展的页数。
File	返回数据库根文件的文件名，包括路径。此属性具有一些在查询属性值可以指定的扩展项。请参见“ DB_EXTENDED_PROPERTY 函数 [System] ”一节《 SQL Anywhere 服务器 - SQL 参考 》。

属性	说明
FileSize	返回系统 dbspace 的文件大小，以页为单位。此属性具有一些在查询属性值时可以指定的扩展项。请参见“ DB_EXTENDED_PROPERTY 函数 [System] ”一节《 SQL Anywhere 服务器 - SQL 参考 》。
FreePages	返回系统 dbspace 中的空闲页数。FreePages 属性仅在使用 8.0.0 或更高版本创建的数据库上受支持。 此属性具有一些在查询属性值时可以指定的扩展项。请参见“ DB_EXTENDED_PROPERTY 函数 [System] ”一节《 SQL Anywhere 服务器 - SQL 参考 》。
FullCompare	返回已执行的超过索引中散列值的比较次数。
GetData	返回 GETDATA 请求数。
GlobalDBID	返回用于在复制环境中生成唯一主键值的 global_database_id 选项的值。
HasCollationTailoring	返回一个值，指示在创建数据库时是否指定归类定制。返回值可能为 On 或 Off。
HasEndianSwapFix	返回一个值，指示数据库是否可以忽略平台的字节排序方式而在所有平台上同时支持 BIG-ENDIAN 和 LITTLE-ENDIAN UTF-16 编码。返回值可能为 On 或 Off。
HashForcedPartitions	返回因内存争用而强制散列运算符进行分区的次数。
HashRowsFiltered	返回位矢量过滤器拒绝的探测行数。
HashRowsPartitioned	返回写入散列工作表的行数。
HashWorkTables	返回为基于散列的操作而创建的工作表数。
HasNCHARLegacyCollationFix	返回以下各值之一： <ul style="list-style-type: none"> ● ON 对应于使用版本 11 或更高版本数据库服务器创建的所有数据库，以及由具有传统归类修复工具的版本 10 数据库服务器创建的、使用传统 NCHAR 归类的数据库。 ● OFF 对应于由不具有传统归类修复工具的版本 10 数据库服务器创建的数据库，或由版本 10 数据库服务器创建的、不使用传统 NCHAR 归类的数据库。
IdentitySignature	保留。

属性	说明
IdleCheck	返回服务器的空闲线程为了执行空闲写入、空闲检查点等操作而变为活动的次数。
IdleChkpt	返回服务器的空闲线程所完成的检查点数。只要空闲线程在高速缓存中写出最后的脏页，即会出现空闲检查点。
IdleChkTime	返回在空闲 I/O 期间执行检查点操作所花费的百分之一秒数。
IdleWrite	返回由服务器的空闲线程所发出的磁盘写入次数。
IndAdd	返回已添加到索引的条目数。
IndLookup	返回已在索引中查找的条目数。
IOParallelism	返回 dbspace 支持的并发 I/O 操作数估计值。此属性具有一些在查询属性值时可以指定的扩展项。请参见“ DB_EXTENDED_PROPERTY 函数 [System] ”一节《 SQL Anywhere 服务器 - SQL 参考 》。
IOToRecover	返回恢复数据库所需的 I/O 操作估计次数。
JavaVM	返回数据库服务器在数据库中执行 Java 时所使用的 Java VM。
Language	返回已知被数据库归类支持的语言的列表（用逗号分隔）。这些语言用双字母 ISO 格式表示。如果语言未知，则返回值为 NULL。有关双字母 ISO 格式语言名称及其所对应语言的列表，请参见“ 了解区域设置语言 ”一节第 389 页。
LockCount	返回数据库所持有的锁的个数。
LockTablePages	返回用于存储锁信息的页数。
LogFileFragments	返回日志文件碎片数。在 Windows 上支持此属性。
LogFreeCommit	返回重做无偿提交数。当请求提交事务日志，但该日志已被写入时，会发生 [重做无偿提交]（这样，提交可以 [无偿] 执行）。
LogMirrorName	返回事务日志镜像的文件名，包括路径。
LogName	返回事务日志的文件名，包括路径。
LogWrite	返回已写入事务日志的页数。
LTMGeneration	返回 LTM 或复制代理的世代号。此属性主要供技术支持使用。
LTMTrunc	返回复制代理的最小确认日志偏移量。

属性	说明
MaxIO	返回 CurrIO 已达到的最大值。
MaxRead	返回 CurrRead 已达到的最大值。
MaxWrite	返回 CurrWrite 已达到的最大值。
MirrorMode	如果数据库镜像未使用，则返回 NULL；如果用 -xp 命令行选项所指定的镜像模式为同步则返回 synchronous，否则返回 asynchronous。
MirrorState	<p>返回以下各值之一：</p> <ul style="list-style-type: none"> ● null 连接到某个没有镜像的数据库。 ● synchronizing 未连接镜像服务器或尚未读取主服务器的所有日志页。如果同步模式为异步，也会返回此值。 ● synchronized 镜像服务器已连接并进行了主服务器上已提交的所有更改。 <p>请参见“数据库镜像简介”一节第 880 页。</p>
MultiByteCharSet	如果数据库使用多字节字符集，则返回 On。否则，返回 Off。
Name	返回数据库名称（与 Alias 相同）。
NcharCharSet	返回数据库的 NCHAR 字符集。
NcharCollation	返回用于 NCHAR 数据的归类名。此属性具有一些在查询属性值时可以指定的扩展项。请参见“DB_EXTENDED_PROPERTY 函数 [System]”一节《SQL Anywhere 服务器 - SQL 参考》。
NextScheduleTime	返回指定事件的下一次计划执行时间；使用 DB_EXTENDED_PROPERTY 函数查询此属性。请参见“DB_EXTENDED_PROPERTY 函数 [System]”一节《SQL Anywhere 服务器 - SQL 参考》。
OptionWatchAction	返回在尝试设置包括在 OptionWatchList 属性中的数据库选项时所采取的操作。请参见“sa_server_option 系统过程”一节《SQL Anywhere 服务器 - SQL 参考》。
OptionWatchList	返回数据库服务器正在监控的数据库选项列表。请参见“sa_server_option 系统过程”一节《SQL Anywhere 服务器 - SQL 参考》。
PageRelocations	返回已从临时文件读取的可重定位堆页数。

属性	说明
PageSize	返回数据库的页大小（以字节为单位）。
PartnerState	返回以下各值之一： <ul style="list-style-type: none"> ● NULL 连接到某个没有镜像的数据库。 ● 已连接 镜像服务器已连接到主服务器。 ● 断开连接 镜像服务器未连接到主服务器。 <p>请参见“数据库镜像简介”一节第 880 页。</p>
Prepares	返回为数据库所执行的语句准备次数。
ProcedurePages	返回已用于过程的可重定位堆页数。
ProcedureProfiling	如果为数据库开启了过程分析，则返回 On。否则，返回 Off。
QueryBypassed	返回从计划高速缓存中重用的请求数。
QueryBypassedCosted	返回优化程序旁路使用开销计算所处理的请求数。
QueryBypassedHeuristic	返回优化程序旁路使用试探法所处理的请求数。
QueryBypassedOptimized	返回最初由优化程序旁路处理、随后由 SQL Anywhere 优化程序完全优化的请求数。
QueryCachedPlans	返回在所有连接间高速缓存的执行计划数。
QueryCachePages	返回用于高速缓存执行计划的页数。
QueryDescribedBypass	返回优化程序旁路所处理的描述请求数。
QueryDescribedOptimizer	返回优化程序所处理的描述请求数。
QueryJHToJNLOptUsed	返回散列连接转换为嵌套循环连接的次数。
QueryLowMemoryStrategy	返回服务器由于内存不足的情况而在执行期间改变其执行计划的次数。由于可用内存小于优化程序估计，或执行计划需要的内存大于优化程序估计，策略可能会改变。
QueryOpened	返回要执行的 OPEN 请求数。
QueryOptimized	返回经过完全优化的请求数。
QueryReused	返回重用查询计划数。
QueryRowsBufferFetch	返回使用缓冲获取的行数。

属性	说明
QueryRowsMaterialized	返回查询处理期间写入工作表的行数。
ReadOnly	如果数据库以只读模式运行，则返回 On。否则，返回 Off。
ReceivingTracingFrom	返回跟踪数据所来自的数据库的名称。如果未附加跟踪，则返回一个空白字符串。
RecoveryUrgency	返回数据库恢复时间设置与恢复数据库所需时间长度的估计值的百分比。请参见“-gr 服务器选项”一节第 184 页和“数据库服务器如何确定何时进行检查点操作”一节第 853 页。
RecursiveIterations	返回递归联合的迭代次数。
RecursiveIterationsHash	返回递归散列连接使用散列策略的次数。
RecursiveIterationsNested	返回递归散列连接使用嵌套循环策略的次数。
RecursiveJNLMisses	返回递归散列连接的索引探测高速缓存未命中次数。
RecursiveJNLProbes	返回递归散列连接尝试索引探测的次数。
RelocatableHeapPages	返回用于可重定位堆（游标、语句、过程、触发器、视图等）的页数。
RemoteTrunc	返回 SQL Remote 消息代理的最小确认日志偏移量。
RollbackLogPages	返回回退日志中的页数。
SendingTracingTo	返回正在向其发送跟踪数据的连接字符串。如果未附加跟踪，则返回一个空白字符串。
SnapshotCount	返回与数据库相关联的快照数。
SnapshotIsolationState	<p>返回以下各值之一：</p> <ul style="list-style-type: none"> ● On 为数据库启用快照隔离。 ● Off 为数据库禁用快照隔离。 ● in_transition_to_on 一旦当前事务完成，便会启用快照隔离。 ● in_transition_to_off 一旦当前事务完成，便会禁用快照隔离。 <p>请参见“allow_snapshot_isolation 选项 [数据库]”一节第 478 页。</p>

属性	说明
SortMergePasses	返回在排序过程中使用的合并遍数。
SortRowsMaterialized	返回写入排序工作表的行数。
SortRunsWritten	返回在排序过程中写入的排序运行数。
SortSortedRuns	返回在运行形成过程中创建的排序运行数。
SortWorkTables	返回为排序创建的工作表数。
StatementDescribes	返回各个 DESCRIBE 请求所处理的语句总数。
StatementPostAnnotates	返回由语义查询转换阶段所处理的语句数。
StatementPostAnnotatesSimple	返回由语义查询转换阶段处理但跳过了某些语义转换的语句数。
StatementPostAnnotatesSkipped	返回已完全跳过语义查询转换阶段的语句数。
SyncTrunc	返回 MobiLink 客户端 dbmlsync 可执行文件的最小确认日志偏移量。
TempFileName	返回数据库临时文件的文件名，包括路径。
TempTablePages	返回用于临时表的临时文件中的页数。
TriggerPages	返回用于触发器的可重定位堆页数。
VersionStorePages	当启用了快照隔离时，返回临时文件中正在用于行版本存储的页数。
ViewPages	返回用于视图的可重定位堆页数。
XPathCompiles	返回数据库服务器自启动以来对任何 XPath 查询（使用 openxml 过程）所进行的编译次数。

物理限制

目录

SQL Anywhere 大小和数量限制	614
----------------------------	-----

SQL Anywhere 大小和数量限制

下表列出了对 SQL Anywhere 数据库中对象的大小和数量的物理限制。在大多数情况下，计算机的内存、CPU 和磁盘驱动器是主要限制因素。

项	限制
数据库大小	每个数据库最多可以包含 13 个文件。对于每个文件，其大小不超过操作系统和文件系统所允许的最大文件的大小
Dbospace 大小	2 ²⁸ x 页面大小
临时文件大小	2 ²⁸ x 页面大小
字段大小	2 GB
文件大小 (FAT 12)	16 MB
文件大小 (FAT 16)	2 GB
文件大小 (FAT 32)	4 GB
文件大小 (NTFS、HP-UX 11.0 及更高版本、Solaris 2.6 及更高版本、Linux 2.4 及更高版本)	<ul style="list-style-type: none"> ● 512 GB (2 KB 页) ● 1 TB (4 KB 页) ● 2 TB (8 KB 页)
文件大小 (所有其它平台和文件系统)	2 GB
最大高速缓存大小 (非 AWE 高速缓存) (Windows 2000 Professional、Windows 2000 Server、Windows XP Home Edition、Windows XP Professional、Windows Server 2003 Web Edition、Windows Server 2003 Standard Edition)	1.8 GB
最大高速缓存大小 (非 AWE 高速缓存) (Windows 2000 Advanced Server、Windows 2000 Enterprise Server、Windows 2000 Datacenter Server、Windows Server 2003 Enterprise Edition、Windows Server 2003 Datacenter Edition、Windows Vista Ultimate、Windows Vista Enterprise、Windows Vista Business、Windows Vista Home Premium、Windows Vista Home Basic)	2.7 GB

项	限制
最大高速缓存大小 (AWE 高速缓存) (Windows 2000 Professional、Windows 2000 Server、Windows 2000 Advanced Server、Windows 2000 Datacenter Server、Windows XP Home Edition、Windows XP Professional、Windows Server 2003 Web Edition、Windows Server 2003 Standard Edition、Windows Server 2003 Enterprise Edition、Windows Server 2003 Datacenter Edition)	100% 的全部可用内存 - 128 MB
最大高速缓存大小 (Windows Mobile)	受设备上可用内存的限制
最大高速缓存大小 (Unix—Solaris、x86 Linux、AIX、HP)	2 GB (32 位服务器)
最大高速缓存大小 (Win 64)	受 64 位服务器上物理内存的限制
最大高速缓存大小 (Itanium HP-UX)	在 64 位服务器上受物理内存的限制
最大索引条目大小	无限制
每台服务器的数据库数	255
每个表的列数	45000 注意: 虽然允许存在过多的列, 但会影响性能。
每个表中可为空的常量数	$\min(45000, (\text{页面大小} - \text{开销}) * 8)$
过程结果集中的列数	45000
SELECT 列表中的列数	100000
GROUP BY 列表中的列数	100000
具有分组集的 GROUP BY 列表中的列数	64
CUBE 中的列数	15
不同的分组集数	32768
列的 DEFAULT 表达式的长度	32768
列的 COMPUTE 表达式的长度	32768
过程参数的 DEFAULT 表达式的长度	32768

项	限制
用户定义域的 DEFAULT 表达式的长度	32768
检查约束大小	2 GB
每个表的索引数	2^{32}
每个数据库的行数	$4096 \times 2^{28} \times 13$
每个表的行数	4096×2^{28}
每个数据库的表数	$2^{32} - 2^{20} - 1 = 4293918719$
每个连接的临时表数	$2^{20} = 1048576$
每个事务的引用表数	无限制
每个数据库的存储过程数	$2^{32} - 1 = 4294967295$
每个数据库服务器的并发语句数	$20 \times \text{number-of-database-connections} + 65534$
每个数据库的事件数	$2^{31} - 1 = 2147483647$
每个数据库的触发器数	$2^{32} - 1 = 4294967295$
行大小	受文件大小的限制
表大小	最大文件大小。表的用户创建索引可以与该表分别进行存储
字符串	2 GB
二进制数据类型	2 GB
标识符（包括用户 ID、表名和列名）	128 个字节
口令	255 个字节
数据库服务器名	250 个字节（TCP/IP 和共享内存） 请参见“-n 服务器选项”一节第 195 页和“ServerName 连接参数 [ENG]”一节第 280 页。
数据库名	250 个字节 请参见“-n 数据库选项”一节第 239 页。

管理数据库

本节介绍如何使用 SQL Anywhere 附带的工具来管理您的数据库。

SQL Anywhere 图形管理工具	619
数据库管理实用程序	693

SQL Anywhere 图形管理工具

目录

使用 Sybase Central	620
使用 Interactive SQL	636
使用文本完成	683
使用快速启动程序选项	686
使用 SQL Anywhere 控制台实用程序	687
检查软件更新	690

使用 Sybase Central

Sybase Central 是用于管理数据库服务器、数据库及它们所包含的对象的图形化工具。

可从 Sybase Central 内选择 [帮助] » [Sybase Central] 来获取有关使用和配置 Sybase Central 的附加信息。

Sybase Central 的主要功能

- **简单的命令访问** 当您选择某个对象时，Sybase Central 中的 [文件] 菜单会自动更新，提供与该对象直接相关的命令。还可以右击对象来访问这些命令。
- **任务向导** 如果您要添加新对象，Sybase Central 将为您提供向导，该向导会引导您逐步完成整个任务。
- **拖放功能** Sybase Central 对于许多操作均支持拖放功能。例如，如果要复制表到不同的数据库，可以单击它们并将其拖到该位置。请参见“在 SQL Anywhere 插件中复制数据库对象”一节第 631 页。
- **键盘快捷方式** 许多常用命令都具有键盘快捷方式；这些快捷方式在菜单中列于命令名称的旁边。请参见“Sybase Central 键盘快捷方式”一节第 625 页。
- **插件支持** 通过使用插件，您可以管理多种数据库产品和工具。在 Sybase Central 中，您可以通过在 [帮助] 菜单中选择插件名来获取使用及配置插件的详细信息。

插件

每个产品均由单独的插件进行管理。必须先注册并装载这些产品的插件，才能在 Sybase Central 中使用这些产品。安装某产品时，会自动注册并装载其插件。

SQL Anywhere 11 包括以下产品的 Sybase Central 插件：

- SQL Anywhere 数据库
- UltraLite 数据库
- MobiLink 同步
- QAnywhere 消息传递

插件文件存在于 SQL Anywhere 11 安装中的以下位置：

插件	文件名和位置
SQL Anywhere 11	<i>install-dir\java\saplugin.jar</i>
MobiLink 11	<i>install-dir\java\mlplugin.jar</i>
UltraLite 11	<i>install-dir\java\ulplugin.jar</i>
QAnywhere 11	<i>install-dir\java\qaplugin.jar</i>

有关使用 SQL Anywhere 11 附带插件的详细信息，请参见：

- SQL Anywhere: “使用 SQL Anywhere 插件”一节第 630 页
- MobiLink: “MobiLink 模型”《MobiLink - 入门》
- UltraLite: “使用 [创建数据库向导] 创建数据库”一节《UltraLite - 数据库管理和参考》和“使用 UltraLite 数据库”《UltraLite - 数据库管理和参考》
- QAnywhere: “QAnywhere 插件”一节《QAnywhere》

部署 Sybase Central

在满足授权协议的前提下，您可以部署 SQL Anywhere 管理工具，包括 Sybase Central。

有关使用应用程序部署 Sybase Central 的详细信息，请参见“部署管理工具”一节《SQL Anywhere 服务器 - 编程》。

启动 Sybase Central

本节介绍了启动 Sybase Central 及使用 SQL Anywhere 插件连接到 Windows 和 Unix 上的示例数据库—SQL Anywhere 11 Demo—的步骤。

◆ 启动 Sybase Central 并连接到示例数据库 (Windows)

1. 从 [开始] 菜单中选择 [程序] » [SQL Anywhere 11] » [Sybase Central]。
2. 在 [欢迎使用 Sybase Central] 窗口中，单击 [查看并编辑数据库上的模式或执行数据库上的维护]。
如果没有出现 [欢迎使用 Sybase Central] 窗口，请选择 [连接] » [使用 SQL Anywhere 11 连接]。
3. 在 [标识] 选项卡中，选择 [ODBC 数据源名称]，然后下方的框中键入 SQL Anywhere 11 Demo。
4. 单击 [确定] 进行连接。

Mac OS X 注释

管理工具只能在使用 Apple JDK 1.6 所支持的 64 位处理器的 Intel Macintosh 机（Mac OS X 10.5.2 或更高）上运行。请参见 <http://www.sybase.com/detail?id=1062623>。

◆ 启动 Sybase Central (Mac OS X)

1. 在 Finder 中，双击 */Applications/SQLAnywhere11* 中的 [Sybase Central]。
2. 在 [欢迎使用 Sybase Central] 窗口中，单击 [查看并编辑数据库上的模式或执行数据库上的维护]。
如果没有出现 [欢迎使用 Sybase Central] 窗口，请选择 [连接] » [使用 SQL Anywhere 11 连接]。
3. 在 [标识] 选项卡中，选择 [ODBC 数据源名称]，然后下方的框中键入 SQL Anywhere 11 Demo。

4. 单击 **[确定]** 进行连接。

注意

以下步骤假定您已发起了运行 SQL Anywhere 实用程序的所需的脚本。请参见“[在 Unix 和 Mac OS X 上设置环境变量](#)”一节第 344 页。

◆ 启动 Sybase Central 并连接到示例数据库（Unix 命令行）

1. 在终端会话中输入以下命令：

```
scjview
```

将打开 Sybase Central。

2. 在 **[欢迎使用 Sybase Central]** 窗口中，单击 **[查看并编辑数据库上的模式或执行数据库上的维护]**。

如果没有出现 **[欢迎使用 Sybase Central]** 窗口，请选择 **[连接]** » **[使用 SQL Anywhere 11 连接]**。

3. 在 **[标识]** 选项卡中，选择 **[ODBC 数据源名称]**，然后键入 **SQL Anywhere 11 Demo**。

如果使用的是支持 Linux 应用程序菜单的 Linux 版本并且选择了在安装 SQL Anywhere 11 时安装菜单项，则可使用以下步骤。

◆ 启动 Sybase Central 并连接到示例数据库（Linux 应用程序菜单）

1. 从 **[应用程序]** 菜单中选择 **[SQL Anywhere 11]** » **[Sybase Central]**。

将打开 Sybase Central。

2. 在 **[欢迎使用 Sybase Central]** 窗口中，单击 **[查看并编辑数据库上的模式或执行数据库上的维护]**。

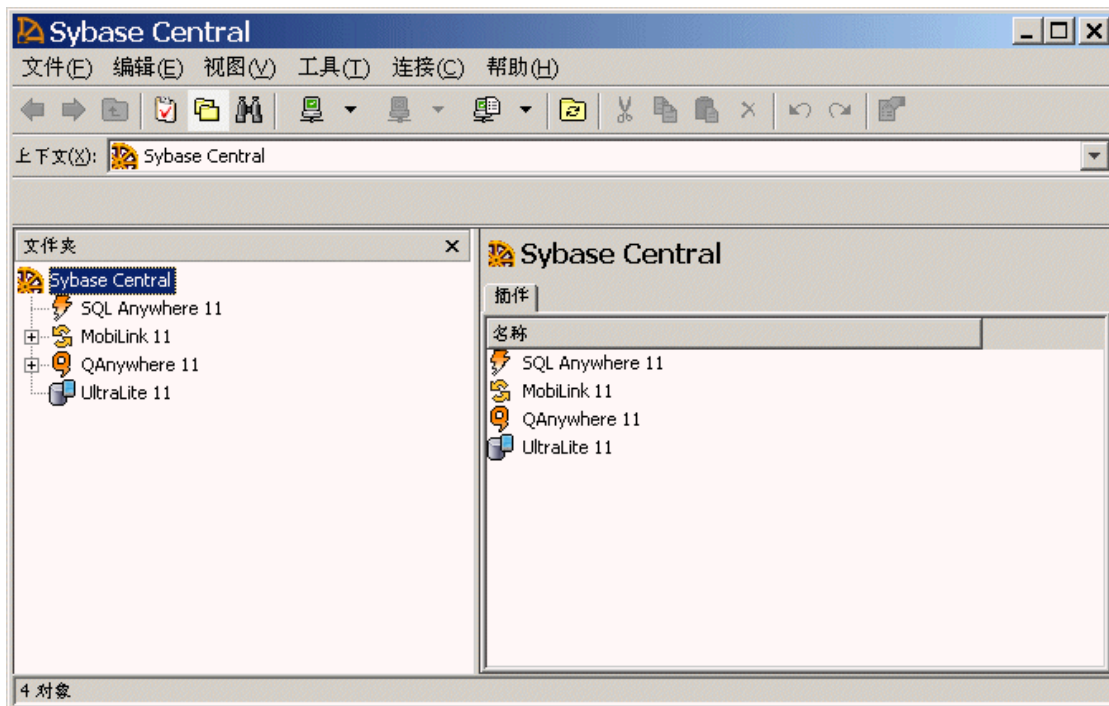
如果没有出现 **[欢迎使用 Sybase Central]** 窗口，请选择 **[连接]** » **[使用 SQL Anywhere 11 连接]**。

3. 在 **[标识]** 选项卡中，选择 **[ODBC 数据源名称]**，然后键入 **SQL Anywhere 11 Demo**。

浏览 Sybase Central

本节介绍如何浏览 Sybase Central 用户界面。

Sybase Central 主窗口：



Sybase Central 主窗口分为两个垂直对齐的窗格。

左窗格

您可以选择是否希望左窗格显示以下内容：

- **[文件夹] 窗格** 显示数据库对象的分层视图。

[文件夹] 只显示对象树中的容器；它不显示不是其它对象的容器的对象。例如，左窗格可能显示一个 [列] 文件夹（一个容器），但不显示列本身，因为这些列是项，应该显示在右窗格中。

- **[任务] 窗格** 显示当前选定数据库对象的任务列表。
- **[搜索] 窗格** 用于在插件中搜索对象。

右窗格

右窗格显示当前选定容器的内容。右窗格中的选项卡显示左窗格中选定容器的内容，以及有关所选容器的其它信息。

通过选择 [视图] » [选择列]，可以配置右窗格中某选项卡上出现的列。

◆ 查看 [任务]、[文件夹] 或 [搜索] 窗格

1. 启动 Sybase Central。
2. 从 [视图] 菜单中，选择 [任务]、[文件夹] 或 [搜索] 以分别查看任务列表、文件夹列表或搜索功能。

可以在 [选项] 窗口中更改右窗格的外观（可通过 [工具] 菜单访问该窗口）。

连接到数据库或数据库服务器之后，您可以通过在主窗口中浏览并选择其对象来管理数据库或数据库服务器。

工具栏

主窗口工具栏提供了常用命令的按钮。要显示或隐藏工具栏，从 [视图] 菜单中，选择 [工具栏] » [标准工具栏]。使用主工具栏可执行以下操作：

- 浏览对象文件夹
- 连接到数据库、数据库服务器或产品插件或者断开与它们的连接
- 显示 [任务]、[文件夹] 或 [搜索] 窗格
- 访问 [连接配置文件] 窗口（也可以通过 [工具] 菜单进行访问）
- 刷新当前文件夹的视图
- 剪切、复制、粘贴和删除对象
- 撤消或恢复操作
- 查看所选对象的属性窗口

[上下文] 下拉列表

使用出现在工具栏下方的 [上下文] 下拉列表，可以浏览插件的对象文件夹。

状态栏

状态栏出现在主窗口的底部，在您浏览菜单时，状态栏会显示菜单命令的简明摘要。要显示或隐藏状态栏，请选择 [视图] » [状态栏]。

在 Sybase Central 中搜索数据库

Sybase Central 允许您搜索数据库中的指定数据库对象，或者搜索数据库对象的 SQL 内的字符串。

◆ 搜索指定对象

1. 在 Sybase Central 中，选择 [视图] » [搜索窗格]。
左窗格中将出现 [搜索] 窗格。
2. 配置搜索选项。
3. 单击 [搜索]。
搜索结果将出现在左窗格的 [结果] 窗格中。
4. 选择一个结果并双击它，以在右窗格中打开该结果。

SQL Anywhere 插件允许采用以下方式搜索：

- **[在 SQL 中搜索] (过程、事件、函数和触发器)** 选择此选项可将过程、事件、函数和触发器的 SQL 包括在搜索中。
- **搜索动态属性 (连接、统计、锁定)** 选择此选项可将动态属性 (如连接的用户、SQL Remote 统计信息、表锁和表页使用信息等) 包括在搜索中。

MobiLink 插件允许采用以下方式搜索：

- **在脚本中搜索** 选择此选项可将同步脚本包括在搜索中。

Sybase Central 键盘快捷方式

Sybase Central 提供了以下键盘快捷方式。

功能键	说明
Alt+Enter	打开选定项的属性窗口。
Ctrl+C	将选定内容复制到剪贴板。
Ctrl+V	插入剪贴板内容。
Ctrl+X	剪切选定内容并将其移动到剪贴板。
删除	删除选定内容。
F1	打开 Sybase Central 帮助。
F5	刷新选定文件夹的内容。
F9	打开 [连接配置文件] 窗口。
F11	在装载有多个插件的情况下打开 [连接] 菜单。如果只装载了一个插件，按 F11 键会为该插件打开 [连接] 窗口。
F12	当 Sybase Central 中只有一个连接时断开该连接。如果有多个连接，则按 F12 键会打开 [断开连接] 窗口，可在其中选择要断开的连接。
Shift+F10	打开选定对象的弹出式菜单。

使用代码编辑器

代码编辑器在 Sybase Central 右窗格中作为 **[SQL]** 选项卡出现，在 Sybase Central 中作为单独窗口出现，在 Interactive SQL 中作为 **[SQL 语句]** 窗格出现，可在其中显示、编辑和打印代码及消息。

除了标准的文本编辑功能之外，代码编辑器还提供了以下功能：

- 工具栏和状态栏
- 自动语法突出显示
- 与语言相关的缩进
- 查找和替换文本的功能
- 从文件打开和保存到文件的功能（此功能是否可用取决于您使用的插件）
- 打印代码的功能
- 键入代码时的文本完成

◆ 在单独的窗口中打开代码编辑器

1. 在 Sybase Central 的左侧或右侧窗格中选择一个数据库对象，例如存储过程、视图或触发器。
2. 选择 [文件] » [在新建窗口中编辑]，或按 Ctrl+E。

自定义代码编辑器

可以使用 [选项] 窗口自定义代码编辑器的显示特性。使用此窗口可以更改前景和背景颜色的设置，以及代码编辑器的整体外观。所做的全部更改都会在各会话间一直保持。

◆ 设置在 [SQL] 选项卡上编辑时的代码编辑器设置

1. 从 [文件] 菜单中选择 [自定义编辑器]。
2. 配置各个选项卡上的设置。单击 [确定]。

◆ 设置在单独窗口中编辑时的代码编辑器设置

1. 在代码编辑器中，选择 [工具] » [选项]。
2. 配置各个选项卡上的设置。单击 [确定]。

代码编辑器键盘快捷方式

Sybase Central 为代码编辑器提供了以下键盘快捷方式。

功能键	说明
Alt+F4	关闭代码编辑器（如果是单独的窗口）或关闭 Sybase Central（如果是在 Sybase Central 的右窗格中编辑）。
Backspace	删除选定内容。如果未选择任何内容，按 Backspace 键可删除光标左侧的字符。

功能键	说明
Ctrl+]	将光标移动到匹配的括号。使用此快捷方式可匹配小括号、大括号、中括号和尖括号。
Ctrl+A	选择 [代码编辑器] 窗口中的全部内容。
Ctrl+Backspace	删除光标左边的单词。
Ctrl+C	将选定文本复制到剪贴板。
Ctrl+Delete	删除光标右边的单词。
Ctrl+End	将光标移动到 [代码编辑器] 窗口的底部。
Ctrl+F	如果未在当前窗口中进行过文本搜索，则打开 [查找/替换] 窗口，可在其中搜索并替换指定的文本。否则，将会搜索选定文本的下一个出现位置。
Ctrl+F3	查找当前选定文本的下一个出现位置。
Ctrl+G	打开 [转到] 窗口，可在其中指定要在 [代码编辑器] 窗口内转到的行位置。
Ctrl+Home	将光标移动到 [代码编辑器] 窗口的顶部。
Ctrl+L	删除当前行。
Ctrl+左箭头键	将光标向后移动一个单词。
Ctrl+N	清除 [代码编辑器] 窗口的内容并关闭当前文件（如果有）。不能在 Sybase Central 右窗格的 [SQL] 选项卡中使用此快捷方式。
Ctrl+O	如果代码编辑器以单独窗口形式打开，则可打开文件。不能在 Sybase Central 右窗格的 [SQL] 选项卡中使用此快捷方式。
Ctrl+P	打印 [代码编辑器] 窗口中的内容。可以按下述方式配置打印文本的外观：从 [工具] 菜单中选择 [选项]，然后单击 [打印] 选项卡。
Ctrl+右箭头键	将光标向前移动一个单词。
Ctrl+S	保存 [代码编辑器] 窗口中的内容。
Ctrl+Shift+]	将选择范围扩展到匹配的括号。使用此快捷方式可匹配小括号、大括号、中括号和尖括号。

功能键	说明
Ctrl+Shift+End	将选择范围扩展到代码的末尾。
Ctrl+Shift+F3	查找当前选定文本的上一个出现位置。
Ctrl+Shift+Home	将选择范围扩展到代码的开头。
Ctrl+Shift+L	删除当前行。
Ctrl+Shift+左箭头键	将选择范围向后扩展一个单词。
Ctrl+Shift+右箭头键	将选择范围向前扩展一个单词。
Ctrl+Shift+U	将选定内容更改为大写字符。
Ctrl+Shift+句点 (.)	增加 [代码编辑器] 窗口中选定文本的行缩进。如果未选定任何文本，则会将缩进应用到当前行。
Ctrl+Shift+逗号 (,)	减少 [代码编辑器] 窗口中选定文本的行缩进。如果未选定任何文本，则会将缩进应用到当前行。
Ctrl+U	将选定内容更改为小写字符。
Ctrl+V	将剪贴板内容插入光标当前所在的位置。
Ctrl+X	剪切选定的文本。
Ctrl+Y	恢复最近撤消的操作。
Ctrl+Z	撤消上一个操作。
Ctrl+减号 (-)	<p>添加或删除双连字符 (--) SQL 注释指示符。</p> <p>要将现有文本变成注释，请在 [代码编辑器] 窗口中选择文本，然后按 Ctrl+减号。双连字符（SQL 注释指示符）将添加到包含选定文本的各行的开头。</p> <p>如果未选定任何文本，注释指示符会添加到当前行的开头。</p> <p>要删除注释指示符，请选择文本然后按 Ctrl+减号。</p> <p>请参见“注释”一节 《SQL Anywhere 服务器 - SQL 参考》。</p>

功能键	说明
Ctrl+正斜线 (/)	<p>添加或删除双斜线 (//) SQL 注释指示符。</p> <p>要将现有文本变成注释，请在 [代码编辑器] 窗口中选择文本，然后按 Ctrl+正斜线。双斜线 (SQL 注释指示符) 会添加到包含选定文本的各行的开头。</p> <p>如果未选定任何文本，注释指示符会添加到当前行的开头。</p> <p>要删除注释指示符，请选择文本然后按 Ctrl+正斜线。</p> <p>请参见“注释”一节《SQL Anywhere 服务器 - SQL 参考》。</p>
删除	删除选定内容。
下箭头键	将光标向下移动一行。
End	将光标移动到当前行的末尾。
F3	如果未在当前窗口中进行过文本搜索，则打开 [查找/替换] 窗口，可在其中搜索并替换指定的文本。否则，将会搜索选定文本的下一个出现位置。
Home	将光标移动到当前行的开始位置或当前行上文本的开始位置。
左箭头键	将光标向左移动一个字符。
Page Down	将光标移动到当前页的末尾。
Page Up	将光标移动到当前页的顶部。
右箭头键	将光标向右移动一个字符。
Shift+下箭头键	将选择范围向下扩展一行。
Shift+End	选择当前行。
Shift+F3	如果未选择任何文本，则打开 [查找/替换] 窗口，您可以在其中搜索并替换指定的文本。如果已选择了文本，则查找选定文本的上一个出现位置。
Shift+F10	<p>显示焦点所在区域的弹出式菜单。</p> <p>此键盘快捷方式与右击区域的作用相同。</p>
Shift+Home	将选择范围扩展到当前行上文本的开始位置。

功能键	说明
Shift+左箭头键	将选择范围向当前选定字符的左边扩展一个字符。
Shift+Page Down	将选择范围向下扩展一页。
Shift+Page Up	将选择范围向上扩展一页。
Shift+右箭头键	将选择范围向当前选定字符的右边扩展一个字符。
Shift+上箭头键	将选择范围向上扩展一行。
上箭头键	将光标向上移动一行。

使用日志查看器

日志查看器是 Sybase Central 中的一个窗口，用于显示和存储产品信息。它会显示以下类型的消息：

- **信息** 有关当前会话的基本信息。
- **警告** 有关已发生操作的警告消息。
- **错误** 有关已失败操作的错误消息。

您可以过滤这些消息以只显示某种类型或一定数量的消息，或选择只显示来自特定插件的消息。您也可以将消息保存到一个文件中，或从列表中清除所有消息。

当在 Sybase Central 中工作时，可通过 [工具] 菜单访问日志查看器。

◆ 打开日志查看器

1. 在 Sybase Central 中，选择 [工具] » [日志查看器]。
即会出现 Sybase Central 日志查看器，显示当前消息（如果有消息存在）。
2. 使用 [视图] 菜单配置所记录的消息类型。

使用 SQL Anywhere 插件

可以使用 SQL Anywhere 插件升级现有数据库、创建新数据库以及管理数据库。可以从 [模式] 菜单或通过单击该模式的工具栏按钮选择所需模式。

SQL Anywhere 插件可在以下任意模式下进行操作：

- **[设计] 模式** 当在 [设计] 模式下工作时，您可以创建和修改数据库对象（如表、用户、触发器、索引、远程数据库服务器等等）。还可以将数据添加到表、创建新数据库以及升级现有数据库。
有关可在 [设计] 模式下在 SQL Anywhere 数据库上执行的任务的详细信息，请参见“[使用数据库对象](#)” 《SQL Anywhere 服务器 - SQL 的用法》。

- **[调试] 模式** 当在 [调试] 模式下工作时，可以使用 SQL Anywhere 调试程序帮助您开发 SQL 存储过程、触发器和事件处理程序。

有关使用 [调试] 模式的详细信息，请参见“[调试过程、函数、触发器和事件](#)”《[SQL Anywhere 服务器 - SQL 的用法](#)》。

- **[应用程序分析] 模式** 当在 [应用程序分析] 模式下工作时，可以为数据库配置应用程序分析或诊断跟踪。生成的数据有助于您理解应用程序与数据库的交互方式，也有助于您确定并消除性能问题。

有关使用 [应用程序分析] 模式的详细信息，请参见“[应用程序分析](#)”一节《[SQL Anywhere 服务器 - SQL 的用法](#)》。

另请参见

- “[通过 Sybase Central、Interactive SQL 或 SQL Anywhere 控制台实用程序进行连接](#)”一节第 86 页
- “[使用数据库对象](#)”《[SQL Anywhere 服务器 - SQL 的用法](#)》
- “[使用 Sybase Central 创建 Windows Mobile 数据库](#)”一节第 323 页

在 SQL Anywhere 插件中复制数据库对象

在 Sybase Central 插件中，可以复制现有的数据库对象，然后将其插入同一数据库或完全不同数据库中的其它位置。

要复制对象，请在 Sybase Central 的左窗格中选择对象，然后将其拖到相应的文件夹或容器中；或者，复制对象，然后将其粘贴到相应的文件夹或容器中。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。在同一个数据库中复制对象时，必须将新对象重命名。

还可以将对象粘贴到数据库中的其它对象上。例如，如果将某个表粘贴到某个用户上，就会授予该用户访问该表的权限。

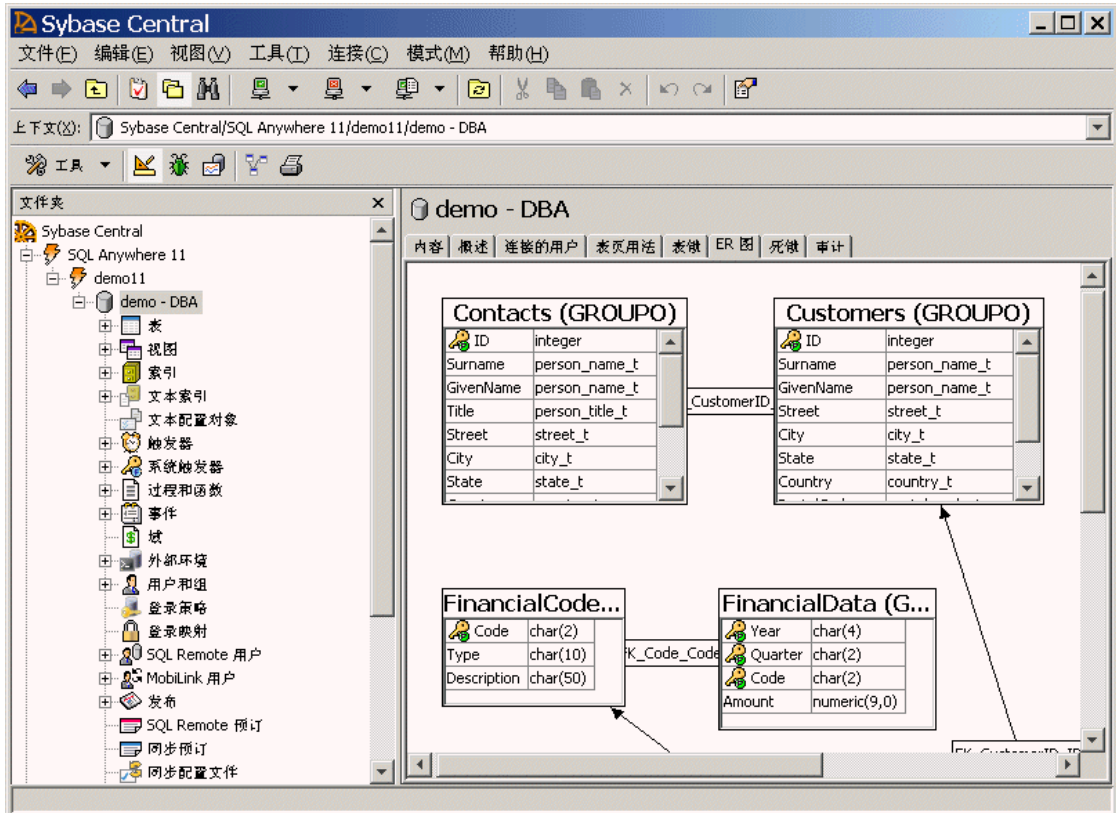
在 Sybase Central 中，当从以下列表复制任何对象时，该对象的 SQL 将被复制到剪贴板上，之后可以将其粘贴到其它应用程序（如 Interactive SQL 或文本编辑器）中。例如，如果在 Sybase Central

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 SQL Anywhere 插件中复制以下对象：

- 项目
- 检查约束
- 列
- Dbspace
- 目录访问服务器
- 域
- 事件
- 外部登录
- 外键
- 索引
- 登录映射（集成登录和 Kerberos 登录）
- 登录策略
- 维护计划报告
- 维护计划
- 消息类型
- MobiLink 用户
- 主键
- 过程和函数
- 发布
- 远程服务器
- 调度
- SQL Remote 预订
- 同步预订
- 系统触发器
- 文本配置对象
- 文本索引
- 表
- 触发器
- 唯一约束
- 用户和组
- 视图
- Web 服务

从 SQL Anywhere 插件查看实体关系图

从 SQL Anywhere 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。

图中出现的表受数据库过滤集制约。过滤是按所有者完成的。

◆ 更改实体关系图中包含的表

1. 请在 Sybase Central 的左窗格中选择数据库，然后选择 [文件] » [配置所有者过滤]。
2. 选择要在实体关系图中看到哪些数据库用户的表，然后单击 [确定]。
3. 选择 [文件] » [按所有者过滤对象]。
4. 单击右窗格中的 [ER 图] 选项卡。
5. 选择 [文件] » [选择 ER 图表]。
6. 在 [选择 ER 图表] 窗口中，使用 [添加] 和 [删除] 按钮即可自定义 [所选表] 列表中出现的表。
7. 单击 [确定]。

另请参见

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

监视数据库健康和统计信息

在 [设计] 模式中，[概述] 选项卡提供了数据库服务器及其功能的概况。此选项卡包含以下组成部分：

- **数据库** 此窗格位于左上角，显示有关数据库服务器的一般信息。

要更新 SQL Anywhere 数据库服务器软件，请单击 [检查更新]。请参见“[检查软件更新](#)”一节第 690 页。

- **功能** 此窗格位于左下角，提供了数据库及其产品和功能的直观表示。单击图中的节点可展开右侧 [健康和统计] 窗格中的补充部分，再次单击节点可折叠该部分。

注意

必须启动 MobiLink 和 QAnywhere 信息的检索；否则这些节点将以未知形式显示（灰显）。请参见下文的“MobiLink、QAnywhere 和通告程序”。

- **健康和统计** 此窗格位于右侧，显示与数据库总体状态相关的统计信息。以下是可用的可折叠窗格：
 - **统计** 显示一般统计信息，如读取和写入磁盘的页数。如果存在任何未调度的请求，则显示一个警告。单击该警告可了解更多信息。
 - **DbSPACE** 显示一个列出所有 dbSPACE 的表。如果 dbSPACE 剩余的可用磁盘空间少于 10%，或者找不到某个 dbSPACE 文件，将显示一个警告。单击该警告可了解更多信息。
 - **事务日志** 显示事务日志的信息，也可显示事务日志镜像的信息（如果适用）。仅当数据库具有事务日志时才会显示此窗格。如果日志文件剩余的可用磁盘空间少于 10%，将显示一个警告。单击该警告可了解更多信息。
 - **连接的用户** 显示连接的用户和事务统计信息。显示一个列出事务次数排前 5 位的事务的表（如果存在）。显示一个列出所有阻塞连接的表（如果存在）。如果存在任何阻塞的连接，则显示一个警告。单击该警告可了解更多信息。
 - **数据库镜像** 显示主服务器、仲裁服务器和镜像服务器的信息以及镜像系统的信息。只有在使用数据库镜像时才会显示此窗格。如果仲裁服务器或镜像服务器断开连接，则显示一个警告。单击该警告可了解更多信息。
 - **远程服务器** 显示列出数据库使用的远程服务器的表。只有远程服务器存在时才会显示此窗格。如果远程服务器已断开连接或远程服务器无法建立连接，则显示一个警告。单击该警告可了解更多信息。

注意

对于 JDBC 远程服务器，只有在 JDBC 远程服务器试图访问代理对象时，才会显示警告提示您连接已断开或丢失。请参见“[有关基于 JDBC 的类的配置说明](#)”一节《SQL Anywhere 服务器 - SQL 的用法》。

- **MobiLink、QAnywhere 和通告程序** 显示 MobiLink、QAnywhere 和通告程序的统计信息。仅当数据库中存在 MobiLink 表和视图时才会显示此窗格。

注意

必须单击 **[刷新]** 按钮刷新此窗格中的信息。与其它窗格中的信息不同，当您选择 **[视图] » [刷新]**（或单击 **[刷新]** 工具栏图标）时不会刷新此窗格中的信息。您必须单独刷新此信息，因为刷新会影响数据库的性能。

- **SQL Remote 用户** 显示一个列出所有 SQL Remote 用户的表，以及他们的最近发送和接收时间。仅当数据库具有 SQL Remote 用户时才会显示此窗格。

记录数据库

您可以使用 **[数据库文档向导]** 在 SQL Anywhere 数据库中生成关于对象的文档。生成的文档包含关于以下数据库对象的信息：

- 过程
- 函数
- 触发器
- 事件
- 视图

除对象定义之外，文档也显示每个对象的依赖性和引用关系。例如，过程 *dbo.sa_migrate_data* 的文档包含它所更新的、插入到的和从中删除对象的表，以及调用该过程的过程的名称。您可以选择在文档中包含对象注释和系统过程。

生成的文档保存为 HTML 文件，以便于浏览和复查。文档对于记录和查看系统状况十分有用。

◆ 生成数据库文档

1. 在 Sybase Central 中，连接到要为其生成文档的数据库。
2. 从 **[工具]** 菜单选择 **[SQL Anywhere 11] » [生成数据库文档]**。
3. 请按照 **[数据库文档向导]** 中的说明进行操作。

使用 Interactive SQL

Interactive SQL 随 SQL Anywhere 提供的一个工具，可使用它来执行 SQL 语句、建立脚本以及显示 SQL Anywhere 数据库和 UltraLite 数据库的数据库数据。可以将 Interactive SQL 用于：

- 将 SQL 语句发送到数据库服务器。请参见“从 Interactive SQL 执行 SQL 语句”一节第 641 页。
- 浏览数据库中的信息。请参见“从 Interactive SQL 执行 SQL 语句”一节第 641 页。
- 编辑结果集中的数据。请参见“在 Interactive SQL 中编辑结果集”一节第 654 页。
- 向数据库中装载数据。请参见“使用 [导入向导] 导入数据”一节《SQL Anywhere 服务器 - SQL 的用法》。
- 将查询结果导出到文件或另一个数据库。请参见“导出查询结果”一节《SQL Anywhere 服务器 - SQL 的用法》。
- 运行脚本文件。请参见“在 Interactive SQL 中运行 SQL 命令文件”一节《SQL Anywhere 服务器 - SQL 的用法》。
- 运行索引顾问，一个有助于提高查询性能的工具。请参见“索引顾问”一节《SQL Anywhere 服务器 - SQL 的用法》。
- 访问查询编辑器，一个有助于设计、分析和测试各种查询的工具。请参见“使用查询编辑器”一节第 649 页。

Interactive SQL 可在 Windows、Solaris、Linux 和 Mac OS X 上使用，请参见 <http://www.sybase.com/detail?id=1062623>。

Mac OS X 注释

管理工具只能在使用 Apple JDK 1.6 所支持的 64 位处理器的 Intel Macintosh 机（Mac OS X 10.5.2 或更高）上运行。请参见 <http://www.sybase.com/detail?id=1062623>。

仅限从 Interactive SQL 执行的 SQL 语句

Interactive SQL 支持所有 SQL Anywhere 和 UltraLite 数据库所支持的 SQL 语句，以及若干只能在 Interactive SQL 中使用的 SQL 语句：

- “CLEAR 语句 [Interactive SQL]” 一节 《SQL Anywhere 服务器 - SQL 参考》
- “CONFIGURE 语句 [Interactive SQL]” 一节 《SQL Anywhere 服务器 - SQL 参考》
- “CONNECT 语句 [ESQL] [Interactive SQL]” 一节 《SQL Anywhere 服务器 - SQL 参考》
- “DESCRIBE 语句 [Interactive SQL]” 一节 《SQL Anywhere 服务器 - SQL 参考》
- “DISCONNECT 语句 [ESQL] [Interactive SQL]” 一节 《SQL Anywhere 服务器 - SQL 参考》
- “EXIT 语句 [Interactive SQL]” 一节 《SQL Anywhere 服务器 - SQL 参考》
- “HELP 语句 [Interactive SQL]” 一节 《SQL Anywhere 服务器 - SQL 参考》
- “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》
- “PARAMETERS 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》
- “SET CONNECTION 语句 [Interactive SQL] [ESQL]” 一节 《SQL Anywhere 服务器 - SQL 参考》
- “SET OPTION 语句 [Interactive SQL]” 一节 《SQL Anywhere 服务器 - SQL 参考》
- “START ENGINE 语句 [Interactive SQL]” 一节 《SQL Anywhere 服务器 - SQL 参考》
- “START LOGGING 语句 [Interactive SQL]” 一节 《SQL Anywhere 服务器 - SQL 参考》
- “STOP LOGGING 语句 [Interactive SQL]” 一节 《SQL Anywhere 服务器 - SQL 参考》
- “SYSTEM 语句 [Interactive SQL]” 一节 《SQL Anywhere 服务器 - SQL 参考》

启动 Interactive SQL

可以采用多种方法来启动 Interactive SQL：从命令提示符、从 Windows 的 [开始] 菜单以及从 Sybase Central 内部。

◆ 启动 Interactive SQL (命令提示符)

- 运行以下命令：

```
dbisql
```

如果未包括 -c 选项（该选项指定了数据库的连接参数）或所提供的连接参数不足，则会出现 [连接] 窗口，可在其中输入数据库的连接信息。请参见“连接参数”一节第 248 页。

要启动 Interactive SQL 并连接到示例数据库，请运行以下命令：

```
dbisql -c "UID=DBA;PWD=sql;DSN=SQL Anywhere 11 Demo"
```

有关所支持选项的信息，请参见“Interactive SQL 实用程序 (dbisql)”一节第 740 页。

◆ 启动 Interactive SQL (Windows)

1. 从 [开始] 菜单中选择 [程序] » [SQL Anywhere 11] » [Interactive SQL]。
2. 在 [连接] 窗口中输入数据库的连接信息。
3. 单击 [确定]。

◆ 启动 Interactive SQL (Sybase Central)

1. 从 [工具] 菜单选择 [SQL Anywhere 11] » [打开 Interactive SQL]。
2. 在 [连接] 窗口中输入数据库的连接信息。
3. 单击 [确定]。

提示

还可以通过以下任意一种方法从 Sybase Central 中访问 Interactive SQL:

- 选择数据库，然后从 [文件] 菜单中选择 [打开 Interactive SQL]。
- 右击数据库，然后选择 [打开 Interactive SQL]。
- 右击存储过程，然后选择 [从 Interactive SQL 执行]。即会打开 Interactive SQL（且在 [SQL 语句] 窗格中含有过程调用），并执行存储过程。
- 右击某个表或视图，然后选择 [在 Interactive SQL 中查看数据]。即会打开含有 `SELECT * FROM table-name` 的 Interactive SQL，并执行查询。

在 Unix 上启动 Interactive SQL

◆ 启动 Interactive SQL (Unix 命令行)

1. 在终端会话中运行以下命令：

```
dbisql
```

2. 在 [连接] 窗口中输入数据库的连接信息。
3. 单击 [确定]。

◆ 启动 Interactive SQL (Mac OS X)

1. 在 Finder 中，双击 `/Applications/SQLAnywhere11` 中的 [Interactive SQL]。
2. 在 [连接] 窗口中输入数据库的连接信息。

如果使用的是支持 Linux 桌面图标的 Linux 版本并且选择了在安装 SQL Anywhere 11 时安装它们，则可使用以下步骤。

◆ 启动 Interactive SQL (Linux 桌面图标)

1. 从 [应用程序] 菜单中选择 [SQL Anywhere 11] » [Interactive SQL]
2. 在 [连接] 窗口中输入数据库的连接信息。
3. 单击 [确定]。

注意

以下步骤假定您已发起了运行 SQL Anywhere 实用程序的所需的脚本。请参见“[在 Unix 和 Mac OS X 上设置环境变量](#)”一节第 344 页。

浏览 Interactive SQL

Interactive SQL 窗口分为以下窗格：

- **SQL 语句** 此窗格为您键入用于访问和修改数据的 SQL 语句提供了场所。

[SQL 语句] 窗格左侧包含一个显示行号的列。这些行号可用于执行以下操作：

- **选择一行** 单击行号以选择一行。或者将光标置于行内，然后按 Ctrl+逗号(,)。
- **选择多行** 单击并拖动以选择多行。
- **选择语句** 双击行号以选择该行对应的整个 SQL 语句。或者将光标置于语句内，然后按 Ctrl+句点(.)。

请参见“[Interactive SQL 键盘快捷方式](#)”一节第 663 页。

- **结果** [结果] 窗格具有两个选项卡：[结果] 和 [消息]。这些选项卡出现在 [结果] 窗格的底部。

[结果] 选项卡显示所执行命令的结果。例如，如果使用 SQL 语句在数据库中搜索特定的数据，则上述窗格中的 [结果] 选项卡会显示符合搜索条件的列和行。可以在 [结果] 选项卡上编辑结果集。请参见“[在 Interactive SQL 中编辑结果集](#)”一节第 654 页。

[消息] 选项卡显示数据库服务器针对您在 Interactive SQL 中执行的 SQL 语句所发出的消息。

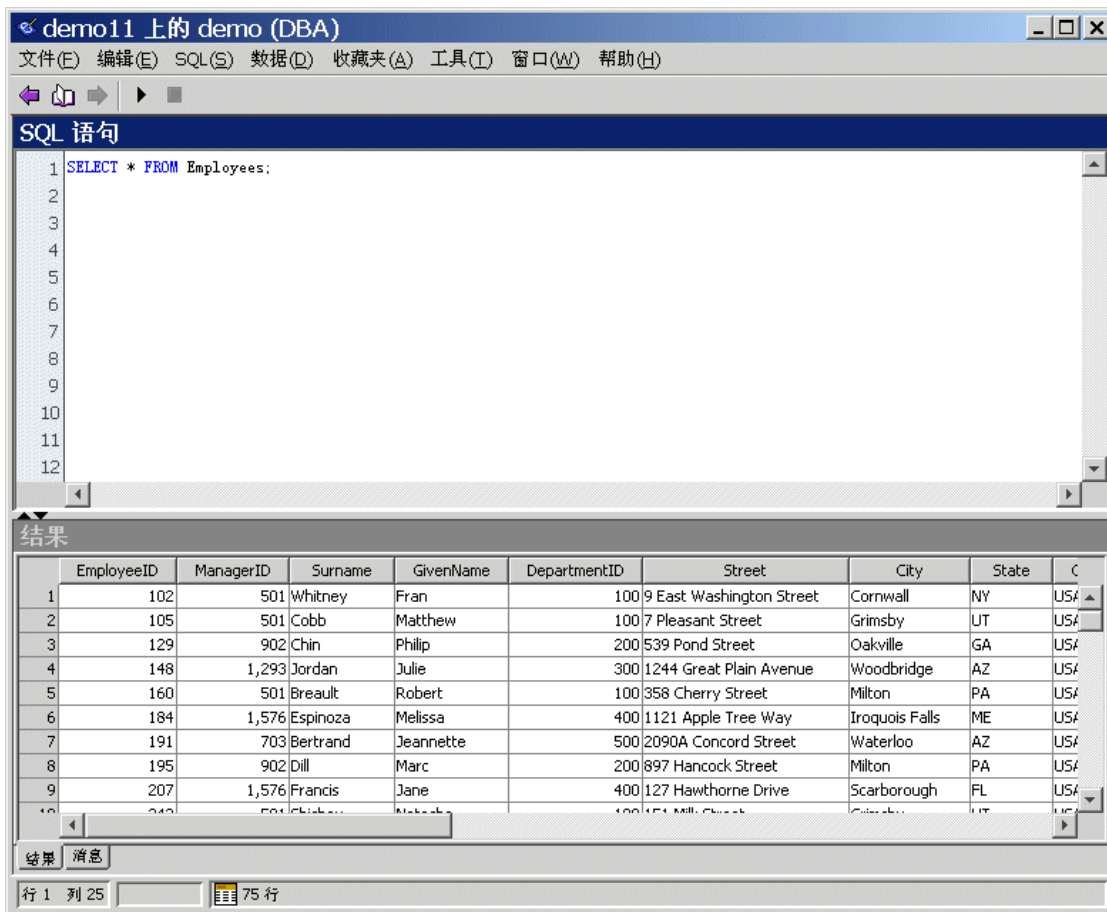
SQL Anywhere 数据库图形式计划和 UltraLite 数据库文本计划的结果显示在各自的 [计划查看器] 窗口中。请参见“[在 Interactive SQL 中查看图形式计划](#)”一节第 652 页。

从 Interactive SQL 连接到数据库后，标题栏会显示连接信息，如下所示：

```
server-name 上的 database-name ( userid )
```

例如，如果使用 SQL Anywhere 11 示例 ODBC 数据源连接到示例数据库，则标题栏包含以下信息：

```
demo ( DBA ) on demoll
```



可以在 Interactive SQL 中使用 [选项] 窗口配置选项卡和窗格的设置。

◆ 自定义 Interactive SQL

1. 在 Interactive SQL 中，选择 [工具] » [选项]。
2. 在左窗格中，单击一个选项并指定需要的选项。
3. 单击 [确定]。

◆ 清除 [SQL 语句] 窗格

- 从 [编辑] 菜单中选择 [清除 SQL] 或按 Esc 键。

另请参见

- “查看数据时排除意外符号问题” 一节第 387 页

Interactive SQL 窗口

您可以通过 [工具]、[数据] 和 [收藏夹] 菜单来访问 Interactive SQL 中的所有窗口。使用这些窗口，可以配置 Interactive SQL 设置、搜索要插入到查询中的表和过程的名称、编辑查询、导出结果集，以及将文件和连接信息作为收藏项保存到收藏夹。

[工具] 菜单包含以下窗口：

- **查寻表名** [查寻表名] 窗口允许您浏览表和列的名称以及将它们插入到 [SQL 语句] 窗格中。
- **查寻过程名** [查寻过程名] 窗口允许您浏览过程的名称以及将它们插入到 [SQL 语句] 窗格中。
- **编辑查询** 查询编辑器提供了一种在 Interactive SQL 中以图形方式创建和编辑 SELECT 语句的方法。请参见“使用查询编辑器”一节第 649 页。
- **索引顾问** [索引顾问] 用于指导您正确地选择索引。可以使用索引顾问分析索引对于某查询的优点。请参见“获取对查询的 [索引顾问] 建议”一节《SQL Anywhere 服务器 - SQL 的用法》。
- **计划查看器** 计划查看器是一个图形工具，用于查看 SQL Anywhere 数据库的图形式计划和 UltraLite 数据库的文本计划。请参见“使用 Interactive SQL 计划查看器查看计划”一节第 651 页。
- **选项** [选项] 窗口可设置 Interactive SQL 中的命令、外观、导入和导出数据以及消息的选项。

[数据] 菜单包含以下窗口：

- **导出** 打开 [导出向导]，可使用该向导导出结果集。请参见“使用 [导出向导] 导出数据”一节《SQL Anywhere 服务器 - SQL 的用法》。
- **导入** 打开 [导入向导]，可以使用该向导从文件或数据库导入数据。请参见“使用 [导入向导] 导入数据”一节《SQL Anywhere 服务器 - SQL 的用法》。

[收藏夹] 菜单包含以下窗口：

- **添加到收藏夹** 此窗口允许您将 SQL 文件和连接信息作为收藏项保存到收藏夹。
- **组织收藏夹** 此窗口允许您维护和组织您的收藏夹。
- **显示收藏夹** 打开 [Interactive SQL] 窗口左侧的 [收藏夹] 窗口。

从 Interactive SQL 执行 SQL 语句

Interactive SQL 的一个主要用途就是浏览表数据。Interactive SQL 通过向数据库服务器发送请求来检索信息。接着，数据库服务器会查找该信息并将结果返回给 Interactive SQL。

执行 SELECT 语句后，结果集会出现在 [结果] 窗格的 [结果] 选项卡上。缺省情况下，在结果集的左侧会出现行号。

注意

即使您不会在 Interactive SQL 中修改所查看的表，数据库服务器也会在这些表上创建模式锁。

然而，可以配置 Interactive SQL 在显示结果集时尝试释放其创建的数据库模式锁。要实现此结果，在 Interactive SQL 中选择 [工具] » [选项] » [SQL Anywhere] 并选择 [自动释放数据库锁]。

如果选择了此选项，则在执行返回结果集的语句之后，Interactive SQL 将检查连接是否具有任何未提交的数据库更改。如果没有，则 Interactive SQL 将释放您的模式锁；否则，Interactive SQL 将不会释放您的模式锁。即，如果您有任何未提交的数据库更改，则 Interactive SQL 将不会释放模式锁。

◆ 执行所有 SQL 语句

1. 在 [SQL 语句] 窗格中键入查询。
2. 按 F5 键，或选择 [SQL] » [执行] 来执行语句。

◆ 执行选定的 SQL 语句

1. 在 [SQL 语句] 窗格中键入查询，然后选择该查询。
2. 按 F9 键，或选择 [SQL] » [执行所选内容] 来执行语句。

要分别执行 SQL 语句（例如，调试时），可以使用 [SQL] 菜单中的 [单步]。[单步] 命令执行指定的语句，然后选中要执行的下一条语句。要执行下一条语句，请再次运行 [单步]。

◆ 一次执行一条 SQL 语句

1. 在 [SQL 语句] 窗格中键入查询。
2. 将光标置于要执行的语句中。
3. 从 [SQL] 菜单中选择 [单步]，或按 Shift+F9 执行指定的语句。
SQL 语句执行完成后，将选中下一条 SQL 语句。
4. 要执行选定的 SQL 语句，请按 Shift+F9。
5. 重复上一步骤直到不再有选定的语句需要执行。

配置执行语句工具栏按钮

还可以单击 **执行** 按钮来执行 [SQL 语句] 窗格中的语句。可将此按钮设置为执行所有 SQL 语句或仅执行选定的语句。

◆ 配置执行语句工具栏按钮

1. 从 [工具] 菜单中选择 [选项]。
2. 单击 [工具栏]
要执行所有 SQL 语句，请选择 [执行所有语句]。这是缺省设置。

要仅执行选定的 SQL 语句，请选择 [执行所选语句]。

另请参见

- “配置管理工具”一节 《SQL Anywhere 服务器 - 编程》

执行多条 SQL 语句

只要每个语句均以命令分隔符结尾，就可以从 Interactive SQL 执行多个 SQL 语句。命令分隔符是用 `command_delimiter` 选项设置的，在缺省情况下为分号 (;)。如果不用分号，替代方法是在一行的开头单独输入分隔符 `go`。请参见“`command_delimiter` 选项 [Interactive SQL]”一节第 670 页。

结果处理

缺省情况下，Interactive SQL 显示所执行的最后一条语句的第一个结果集。

◆ 查看所有结果集

1. 从 [工具] 中选择 [选项]，然后选择以下各项之一：
 - [SQL Anywhere] » [结果]。
 - [UltraLite] » [结果]。
2. 选择 [显示所有结果集]。
3. 选择 [显示每条语句的结果]。
4. 单击 [确定]

提示

可按 F9 键只执行 [SQL 语句] 窗格中选定的文本。

可按 Shift+F9 只执行 [SQL 语句] 窗格中选定的语句，并选中下一条要执行的语句。

另请参见

- “查看数据时排除意外符号问题”一节第 387 页

执行命令文件

命令文件是包含 SQL 语句的文本文件，在重复运行同样的 SQL 语句时十分有用。可使用 Interactive SQL 打开、查看、运行和保存命令文件。

还可采用以下任意方式从 Interactive SQL 中执行命令文件：

- 可以使用 Interactive SQL 的 READ 语句执行命令文件。例如，下面的语句执行文件 `temp.sql`：

```
READ temp.sql;
```

- 可以将命令文件装载到 [SQL 语句] 窗格中并在那里直接执行此命令文件。

通过选择 [文件] » [打开] 将命令文件装载到 [SQL 语句] 窗格中。出现提示时请输入文件名，例如 *temp.sql*。

- 通过选择 [文件] » [运行脚本] 可以不装载命令文件即执行它。
- 可以提供一個命令文件作为 Interactive SQL 的命令行参数。

请参见“在 Interactive SQL 中运行 SQL 命令文件”一节《SQL Anywhere 服务器 - SQL 的用法》。

将 Interactive SQL 设置为 .sql 文件的缺省编辑器

在 Windows 平台上，可以将 Interactive SQL 作为 .sql 命令文件的缺省编辑器。这样便可双击文件以使其内容出现在 Interactive SQL 的 [SQL 语句] 窗格中。

◆ 使 Interactive SQL 成为 .sql 文件的缺省编辑器

1. 从 Interactive SQL 中选择 [工具] » [选项]。
2. 在左窗格中，单击 [常规]。
3. 单击 [使 Interactive SQL 成为 .SQL 文件和计划文件的缺省编辑器]。
4. 单击 [确定]。

有关使用 Interactive SQL 处理命令文件的详细信息，请参见：

- “使用 SQL 命令文件”一节《SQL Anywhere 服务器 - SQL 的用法》
- “在 Interactive SQL 中运行 SQL 命令文件”一节《SQL Anywhere 服务器 - SQL 的用法》

使用收藏夹

在 Interactive SQL 中，您可将频繁使用的 SQL 命令文件和连接存储在收藏夹列表中。收藏夹列表是特定于单个用户的，其他用户无法看到。

◆ 将 .sql 文件添加到收藏夹

1. 打开要添加到收藏夹的 SQL 命令文件。
2. 从 [收藏夹] 菜单选择 [添加到收藏夹]。
3. 选择 [添加打开文件 '*filename*']。在 [名称] 字段中，键入 .sql 文件的名称。
4. 单击 [确定]。

◆ 将连接添加到收藏夹

1. 连接到数据库。
2. 从 [收藏夹] 菜单选择 [添加到收藏夹]。
3. 选择 [保存连接口令]。在 [名称] 字段中，键入连接的名称。
4. 单击 [确定]。

可以选择在提要栏中显示收藏夹。

◆ 显示收藏夹

- 从 [收藏夹] 菜单选择 [显示收藏夹]。

[收藏夹] 窗格将出现在 [Interactive SQL] 窗口的左侧。

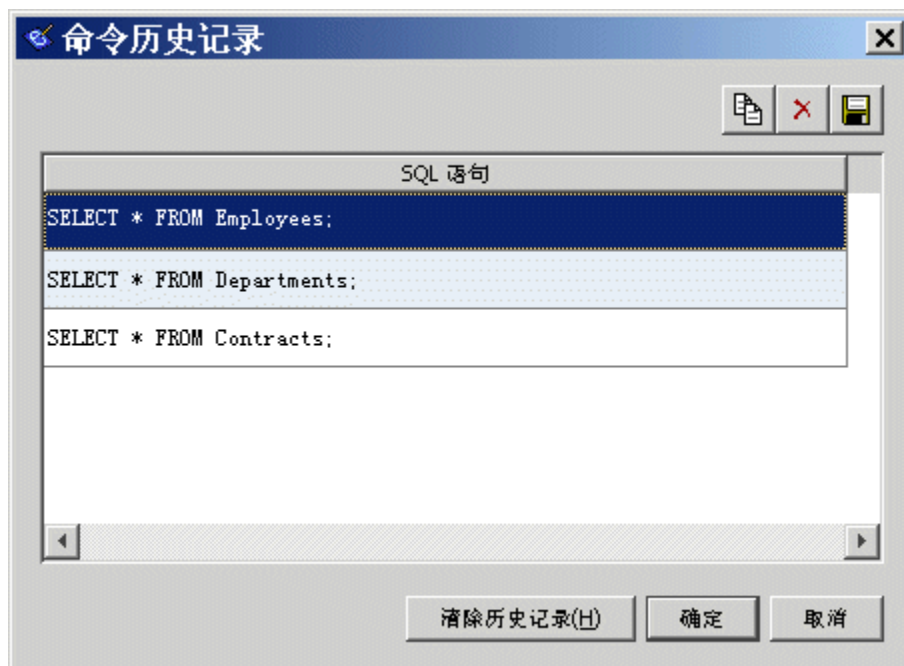
◆ 打开收藏夹

- 从 [收藏夹] 菜单选择要打开的收藏夹。

重新调用命令

当您执行命令时，Interactive SQL 会自动将其保存在一个历史记录列表中，该列表在各 Interactive SQL 会话间一直保持。Interactive SQL 维护的记录包含多达 50 个最近执行的命令。

可在 [命令历史记录] 窗口中查看整个命令列表。要访问 [命令历史记录] 窗口，请按 Ctrl+H，或单击工具栏上的 [打开过去的 SQL 语句的列表] 图标。



最近的命令出现在列表的底部。要重新调用某个命令，请选择该命令，然后单击 [确定]。它将出现在 Interactive SQL 的 [SQL 语句] 窗格中。可从 [命令历史记录] 窗口中选择多个命令。

也可在不使用 [命令历史记录] 窗口的情况下重新调用命令。使用工具栏中的 [重新调用前一 SQL 语句] 和 [重新调用下一 SQL 语句] 图标可向后或向前滚动以选择命令，或者分别按 Alt+右箭头和 Alt+左箭头。

注意

如果执行包含口令信息的 SQL 语句（CREATE USER、GRANT REMOTE DBA、CONNECT 或 CREATE EXTERNLOGIN），该口令信息会出现在 [命令历史记录] 窗口中并保持到当前 Interactive SQL 会话结束。

在后续的 Interactive SQL 会话中查看命令历史记录时，包含口令信息的任何语句中的口令都会被替换为 [...]。例如，如果在 Interactive SQL 中执行以下语句：

```
CREATE USER testuser  
IDENTIFIED BY testpassword;
```

在后续的 Interactive SQL 会话中，[命令历史记录] 窗口中将显示以下语句：

```
CREATE USER testuser  
IDENTIFIED BY ...;
```

从 [命令历史记录] 窗口中复制命令

可从 [命令历史记录] 窗口中复制命令以在别处使用。复制多个命令时，命令以命令分隔符（缺省情况下是分号）进行分隔。

◆ 从 [命令历史记录] 窗口中复制命令

1. 打开 [命令历史记录] 窗口。
2. 选择一个命令或多个命令，然后按 Ctrl+C 或单击 [复制]。
3. 单击 [确定] 将选定的语句复制到 Interactive SQL 的 [SQL 语句] 窗格。

从 [命令历史记录] 窗口中保存命令

还可以将命令保存在文本文件中，以便在随后的 Interactive SQL 会话中使用它们。

◆ 将命令历史记录保存到文件中

1. 打开 [命令历史记录] 窗口。
2. 单击 [将历史记录另存为 .SQL 文件] 按钮或按 Ctrl+S。
3. 在 [另存为] 窗口中，为该文件指定一个位置和名称。
命令历史记录文件的扩展名为 *.sql*。
4. 完成后单击 [保存]。

从 [命令历史记录] 窗口中删除命令

[命令历史记录] 窗口的内容会在各 Interactive SQL 会话间一直保持。可采用以下两种方式之一从历史记录中删除命令：

- 选择一个或多个命令，然后单击 [删除] 按钮或按 Delete 键从窗口中删除所选命令。此操作无法撤销。
- 单击 [清除历史记录] 从窗口中删除所有命令。此操作无法撤销。

记录命令

使用 Interactive SQL 记录功能，可在执行命令时记录这些命令。Interactive SQL 会一直记录命令，直到您停止记录进程或直到您结束当前会话。记录的命令存储在日志文件中，从而可以再次使用这些命令。

◆ 开始记录 Interactive SQL 命令

1. 从 [SQL] 菜单中选择 [开始记录]。
2. 在 [另存为] 窗口中，为该日志文件指定一个位置和名称。例如，将文件命名为 *mylogs.sql*。
3. 完成后单击 [保存]。

◆ 停止记录 Interactive SQL 命令

- 从 [SQL] 菜单中选择 [停止记录]。

提示

也可通过在 [SQL 语句] 窗格中键入的方式来开始和停止记录。要开始记录，请键入并执行 **START LOGGING 'c:\filename.sql'**，其中 *c:\filename.sql* 是日志文件的路径、名称和扩展名。如果路径包含嵌入的空格，只需用单引号括起来即可。要停止记录 Interactive SQL 命令，请键入并执行 **STOP LOGGING**。

开始记录后，便会记录您尝试执行的所有命令，包括那些未正确执行的命令。

在 Interactive SQL 中取消命令

取消操作会停止当前的处理并提示输入下一个命令。Interactive SQL 工具栏上的 [中断 SQL 语句] 按钮可用于取消命令。

如果当时正在处理命令文件，则会提示您选择要采取的操作 ([停止命令文件]、[继续] 或 [退出 Interactive SQL])。可以通过 Interactive SQL on_error 选项来控制这些操作。请参见“on_error 选项 [Interactive SQL]”一节第 679 页。

插入注释

注释用于在 SQL 语句或语句块中附加说明性文字。数据库服务器不执行注释。SQL Anywhere 支持以下类型的注释：--（双连字符）、//（双斜线）和 /* ...*/（斜线加星号）。请参见“注释”一节《SQL Anywhere 服务器 - SQL 参考》。

◆ 添加或删除注释指示符

- 要将现有文本变成注释，请在 [SQL 语句] 窗格中选中文本，然后按 Ctrl+减号 (-) 添加双连字符注释指示符或按 Ctrl+正斜线 (/) 添加双斜线注释指示符。SQL 注释指示符会添加到包含选定文本的各行的开头。

如果未选定任何文本，注释指示符会添加到当前行的开头。

要删除注释指示符，请选中文本，然后按 Ctrl+减号 (-) 删除双连字符注释指示符或按 Ctrl+正斜线 (/) 删除双斜线注释指示符。

缩进 SQL 语句

◆ 添加或增加 SQL 语句的缩进量

1. 在 [SQL 语句] 窗格中选中要进行缩进的文本。如果未选定任何文本，则会将缩进应用到当前行。
2. 按 Ctrl+句点 (.)。

◆ 删除或减少 SQL 语句的缩进量

1. 在 [SQL 语句] 窗格中选中要减少缩进量的文本。如果未选定任何文本，则会将缩进应用到当前行。
2. 按 Ctrl+逗号 (,)。

◆ 更改缩进的空格数

1. 从 [工具] 菜单中选择 [选项]。
2. 选择 [编辑器]，然后单击 [制表符] 选项卡。
3. 在 [缩进大小] 字段中键入新数字。

查找表、列和过程

在 Interactive SQL 中输入命令时，可以查找存储在当前数据库中的表、列或过程的名称并将其插入光标所在的位置。

◆ 在数据库中查找表的名称：

1. 从 [工具] 菜单选择 [查寻表名] 或按 F7 键。
2. 查找并选择该表。
3. 单击 [确定]，将表名插入到 [SQL 语句] 窗格中的当前光标位置。

◆ 在数据库中查找列名称

1. 从 [工具] 菜单选择 [查寻表名] 或按 F7 键。
2. 查找并选择包含该列的表。
3. 单击 [显示列]。
4. 选择该列并单击 [确定]，列名将被插入到 [SQL 语句] 窗格中的当前光标位置。

◆ 在数据库中查找过程的名称

1. 从 [工具] 菜单选择 [查寻过程名] 或按 F8 键。
2. 查找并选择该过程。
3. 单击 [确定]，将过程名插入到 [SQL 语句] 窗格中的当前光标位置。

在 [查寻表名] 和 [查寻过程名] 窗口中，可以输入所要查找的表或过程的前几个字符。列表范围将会缩小到仅包含那些以所输入文本开头的项。

可以使用 SQL 通配符 '%'（百分号）和 '_'（下划线）帮助缩小搜索范围。 '%' 匹配由零个或多个字符组成的任意字符串，而 '_' 匹配任一字符。

例如，要列出包含 profile 一词的所有表，请键入 %profile%。

如果要在表名中搜索百分号或下划线，必须在百分号或下划线前加一个转义字符。iAnywhere JDBC 驱动程序的转义字符是 '\（代字号）。

提示

在 [SQL 语句] 窗格中键入时，Interactive SQL 支持数据库对象名的文本完成，因此也可以改用这种方法来查找表名和过程名。请参见“使用文本完成”一节第 683 页。

还可以使用文本完成来查找对象名，包括表、列和过程。请参见“使用文本完成”一节第 683 页。

从结果集生成 SQL 语句

您可以在结果集中为所选行创建 INSERT、DELETE 和 UPDATE 语句。

◆ 从 Interactive SQL 结果集生成 SQL 语句

1. 选择要为其生成语句的行。
2. 右击所选行，然后选择 [生成]，再选择 [INSERT 语句]、[DELETE 语句] 或 [UPDATE 语句]。语句将被复制到剪贴板。

使用查询编辑器

查询编辑器是 Interactive SQL 中用来帮助构建 SELECT 语句的工具。可以在查询编辑器中创建 SQL 查询，也可以导入查询并对其进行编辑。完成查询后，单击 [确定] 将其导回 Sybase Central 或 Interactive SQL 进行处理。

◆ 使用查询编辑器创建查询

1. 从 Interactive SQL 连接到数据库。
2. 打开查询编辑器。

从 [工具] 菜单中选择 [编辑查询]。

如果在 Interactive SQL 中选择了 SQL 代码，则所选代码将自动导入到查询编辑器中。

3. 创建查询。
4. 单击 [确定] 将查询写入到 Interactive SQL 的 [SQL 语句] 窗格。

查询编辑器提供一系列选项卡，这些选项卡可指导您设置 SQL 查询的组成部分，其中多数组成部分是可选的。下面按建立 SQL 查询时通常采用的顺序介绍这些选项卡：

选项卡	说明
[表] 选项卡	使用该选项卡可指定查询中的表。
[连接] 选项卡	使用该选项卡可指定用于组合表数据的连接策略。如果查询中包括多个表，则应指定用于组合这些表中的数据的连接策略。如果不为 [表] 选项卡中添加的表指定连接策略，查询编辑器会推荐一个；如果表之间存在外键关系，查询编辑器会基于该关系生成连接条件，否则会建议使用矢量积。当打开查询时，查询编辑器将完全接受您已指定的连接策略（未指定的 JOIN 不会像在 SQL Anywhere 中一样缺省设置为 KEY JOIN）。
[列] 选项卡	使用该选项卡可指定结果集中的列。如果不指定列，则将显示所有列。
[INTO] 选项卡	使用该选项卡可将结果赋给变量。
[WHERE] 选项卡	使用该选项卡可指定条件限制结果集中的行。
[GROUP BY] 选项卡	使用该选项卡可分组结果集中的行。
[HAVING] 选项卡	使用该选项卡可基于分组值限制结果集中的行。
[ORDER BY] 选项卡	使用该选项卡可对行进行排序。

查询编辑器还包含下列工具：

窗口	说明
表达式编辑器	使用 [表达式编辑器] 可建立搜索条件或定义计算的列。
派生表	该窗口与主查询编辑器几乎完全相同，使用它可以创建派生表和子查询。

查询编辑器的每个组件都有上下文相关的联机帮助，用于介绍如何使用选项卡，并提供指向解释相关概念和用法的 SQL Anywhere 文档的链接。

有了查询编辑器，不使用 SQL 代码也能创建查询。不过，您可以按以下方法在查询编辑器中使用 SQL：

- 可以在 Interactive SQL 的 [SQL 语句] 窗格中创建查询，然后在打开查询编辑器之前通过突出显示该代码将它导入到编辑器中。
- 使用查询编辑器时，可随时单击窗口底部的 [SQL] 来查看正在建立的查询的 SQL 代码。可以直接编辑该代码，查询编辑器中的字段会自动更新。

您可以通过 Interactive SQL 或 Sybase Central 配置查询编辑器以使 SQL 完全结构化，即所有的表名和列名都已完全限定且名称已加上引号。这些额外的格式设置通常并不需要，但它可以确保 SQL 在所有情况下均能正常工作。您也可以选择在启动时获取表列表。

◆ 配置查询编辑器

- 从 [工具] 菜单中选择 [选项] » [SQL Anywhere]，然后单击 [查询编辑器] 选项卡。

查询编辑器的限制

查询编辑器建立 SQL Anywhere SELECT 语句。它并不是为创建视图而设计的，虽然您可以在 Interactive SQL 中创建视图并在查询编辑器中引用它们。它也不是用来创建 UPDATE 语句或其它非 SELECT SQL 语句的。它创建单个 SELECT 语句，因此不建立 SELECT 语句的联合或交叉。另外，查询编辑器不支持 Transact-SQL 语法。

另请参见

- “查询数据” 《SQL Anywhere 服务器 - SQL 的用法》。
- “SELECT 语句” 一节 《SQL Anywhere 服务器 - SQL 参考》。

使用 Interactive SQL 计划查看器查看计划

计划查看器是一个图形工具，用于查看 SQL Anywhere 数据库的图形式计划和 UltraLite 数据库的文本计划。

◆ 启动计划查看器

1. 打开 Interactive SQL。
2. 选择 [工具] » [计划查看器]（或按 Shift+F5）。

[计划查看器] 即会出现在单独的窗口中。

浏览计划查看器

[计划查看器] 窗口分为以下窗格：

- [SQL] 窗格 此窗格为您键入要为其生成计划的 SQL 语句提供了场所。
- [结果] 窗格 此窗格显示图形式计划。此窗格仅用于 SQL Anywhere 数据库。
- [详细信息] 窗格 此窗格提供有关 SQL Anywhere 数据库计划的文本式详细信息。对于 UltraLite 数据库，此窗格显示文本计划。



在 Interactive SQL 中查看图形式计划

您可以在 Interactive SQL 的 [计划查看器] 窗口中查看查询优化程序针对 SQL 语句的执行计划。

对于 SQL Anywhere 数据库，[计划查看器] 中只显示图形式计划。对于 UltraLite 数据库，[计划查看器] 中只支持文本计划。请参见“查看 UltraLite 执行计划”一节《UltraLite - 数据库管理和参考》。

◆ 创建图形式计划

1. 在 [SQL 语句] 窗格中键入查询。
2. 选择 [工具] » [计划查看器] 或按 Shift+F5。

[计划查看器] 即会出现在单独的窗口中。指定的查询会出现在 [SQL] 窗格中。

3. 单击 [获取计划] 为指定查询生成计划。

◆ 打开图形式计划

1. 选择 [工具] » [计划查看器]。
2. 单击 [打开]。
3. 选择一个计划文件 (.saplan)，然后单击 [打开]。

另请参见

- “读取图形式计划”一节 《SQL Anywhere 服务器 - SQL 的用法》

配置图形式计划

在执行图形式计划后，可以自定义计划中各项的外观。

◆ 更改图形式计划的外观

1. 在 [计划查看器] 左下角的窗格中右击计划，然后选择 [自定义]。
2. 更改设置。
3. 完成后单击 [确定]。
4. 单击 [获取计划] 以根据您的更改生成计划。

打印 SQL 语句、执行计划和结果集

可以通过以下方式打印 [SQL 语句] 窗格或查询结果的内容：

- 按 Ctrl+P
- 选择 [文件] » [打印]

当出现提示时，选择打印 [SQL 语句] 或 [结果]。

可通过以下方式打印 [计划查看器] 中的计划：

- 按 [打印] 按钮
- 右击计划，然后选择 [打印]

还可以在 Interactive SQL 的 [选项] 窗口中添加页眉或页脚和配置其它格式选项。

◆ 添加页眉

1. 打开 Interactive SQL 的 [选项] 窗口。
选择 [工具] » [选项]。
2. 在 [编辑器] 页面上，单击 [打印] 选项卡。

3. 在 [页眉] 字段中，指定想要在页眉中显示的文本。也可以单击右箭头，选择要包括在页眉中的项。

在 Interactive SQL 中编辑结果集

在 Interactive SQL 中执行查询之后，可以排序和编辑结果集来修改数据库。还可以从结果集中选择行并进行复制以供在其它应用程序中使用。结果中的字段分隔符、引号字符和转义字符分别由 `isql_field_separator`、`isql_quote` 和 `isql_escape_character` 选项控制。在 Interactive SQL 的 [选项] 窗口中可以查看和更改这些选项。

Interactive SQL 支持编辑、插入和删除行。编辑结果集与执行 UPDATE、INSERT 和 DELETE 语句等效。在编辑结果集后，等效的 INSERT、UPDATE 和 DELETE 语句将被添加到 Interactive SQL 的命令历史记录。请参见“[重新调用命令](#)”一节第 645 页。

要编辑结果集中的某一行或某个值，您必须对要在其中修改值的表或列具有适当的权限。例如，如果要删除某行，则您必须对该行所属的表具有 DELETE 权限。

在以下情况下，不能编辑结果集：

- 从具有主键的表中选择了列，但没有选择所有的主键列。
- 试图编辑 JOIN 的结果集（例如，结果集中存在来自多个表的数据）。
- 试图编辑已禁用编辑的表，请参见“[禁用表编辑](#)”一节第 655 页。

在以下情况下，编辑结果集可能会失败：

- 试图编辑您不具有权限的行或列。
- 输入了无效值（例如，在数字列中输入字符串或在不允许空值的列中输入空值）。

当编辑失败时，将出现一条解释该错误的 Interactive SQL 错误消息，而数据库表值保持不变。

从 Interactive SQL 结果集中编辑表值

在 Interactive SQL 中，可以更改数据库表中现有行内的任何值或所有值，只要对所修改的列具有 UPDATE 权限。另外，对于 SQL Anywhere 和 UltraLite 数据库，不得禁用表编辑。

编辑结果集时，一次只能对一行中的值进行更改。

◆ 编辑结果集中的行

1. 在 Interactive SQL 中执行查询
2. 在 [结果] 选项卡上，单击要更改的值。
3. 右击该值，然后选择 [编辑行]，或按 F2 键编辑该结果集。
包含该值的表单元格中即会出现一个闪烁的光标。
4. 输入新值。如果要更改该行中的其它值，请按 Tab 键或 Shift+Tab 移动到其它值。

5. 完成对行中值的编辑后，即可按 Enter 键更新数据库。
可以按 Esc 键取消对选定值所做的更改。
6. 执行 COMMIT 语句以使您对表所做的更改成为永久更改。

禁用表编辑

可以通过 Interactive SQL 中的 [选项] 窗口或 Interactive SQL 的初始化文件禁用表编辑，请参见“配置管理工具”一节《SQL Anywhere 服务器 - 编程》。

◆ 禁用表编辑 (Interactive SQL)

1. 从 [工具] 菜单中选择 [选项]，然后选择 [SQL Anywhere] 或 [UltraLite]。
2. 确保选择了 [可滚动表]，然后选择 [禁用编辑]。
3. 单击 [确定]。
4. 执行查询。

为使对表编辑的更改生效，必须执行一次新查询。

从 Interactive SQL 结果集向数据库中插入行

Interactive SQL 允许您向表中添加新行。可使用 Tab 键在结果集的列之间切换以将值添加到相应的行。必须对要添加新行的表具有 INSERT 权限。

◆ 向结果集中插入新行

1. 右击结果集，然后选择 [添加行]。
即会出现一个新的空行，在该行的第一个值中有一个闪烁的光标。
2. 输入新值，然后按 Tab 键移动到下一列。
不能在列中输入无效数据类型。例如，不能将字符串输入到接受 INT 数据类型的列中。
重复此步骤直到添加完所有的列值。
3. 按 Enter 键更新数据库。

向具有缺省值的列中插入值

在具有缺省值的列中添加值时，单元格编辑器包含一个含有 [(缺省值)] 项的列表。如果要插入缺省值，请选择 [(缺省值)]。类似地，如果列接受空值，则列表中会出现 [(NULL)]。如果列不能为空值且没有缺省值，则必须输入一个值。

向计算列中插入值

如果结果集包含计算列且您没有为该计算列指定值，则会在更新数据库时计算该值。但是，如果为计算列指定了值，则会用指定的值更新数据库，而不会为计算列计算值。

使用 INPUT 语句插入新行

除了利用结果集外，您也可以使用带 PROMPT 子句的 INPUT 语句在 Interactive SQL 中插入新行。指定 PROMPT 子句后，Interactive SQL 会提示您为表中的各列提供值。例如，要向 Products 表中添加新行并获得为各列提供值的提示，可以在 Interactive SQL 中执行以下语句：

```
INPUT INTO Products PROMPT;
```

使用 Interactive SQL 从数据库中删除行

您还可以在 Interactive SQL 中从数据库表中删除行。您必须对要删除行的表具有 DELETE 权限。

◆ 从结果集中删除行

1. 选择要删除的行。要选择行，请执行以下操作：
 - 按住 Shift 键的同时单击行。
 - 按住 Shift 键的同时使用上箭头键或下箭头键。如果要删除不连续的行，必须分别删除每行。
2. 按 Delete 键。
即会从数据库表中删除选定的行。
3. 执行 COMMIT 以使更改成为永久更改。

从 Interactive SQL 结果集中复制行

您可以直接从 Interactive SQL 的结果集中复制单元格、行和列，然后将它们粘贴到其它应用程序中。复制行和列时会将列标题和表数据都复制到剪贴板中。一次只能复制一行。

将根据以下 Interactive SQL 选项确定所复制数据的格式：

- [“isql_field_separator 选项 \[Interactive SQL\]”](#) 一节第 676 页
- [“isql_escape_character 选项 \[Interactive SQL\]”](#) 一节第 675 页
- [“isql_quote 选项 \[Interactive SQL\]”](#) 一节第 678 页

您还可以从 Interactive SQL 的 [选项] 菜单中选择 [导入/导出] 来更改这些选项。

如果这些选项均设置为其缺省值，则复制的数据以逗号分隔并且所有的字符串都括在单引号中。

◆ 从 Interactive SQL 结果集中复制行：

1. 选择结果集中要复制的行。
2. 右击所选行，然后选择 [复制] » [复制所选行]。
选定的行（包括其列标题）将会被复制到剪贴板中。
现在便可将其粘贴到其它应用程序中。

◆ 从 Interactive SQL 结果集中复制列

- 右击要复制的列，然后选择 [复制] » [复制列]。一次只能复制一列。

如果 [结果] 窗格未包含整个结果集，则在选择它们之前会提示您获取其余结果。否则，只会选择目前已获取的那些结果。

选定的列（包括列标题）将会被复制到剪贴板中。

现在便可将列粘贴到其它应用程序中。

◆ 从 Interactive SQL 结果集中复制各个值

- 右击结果集中要复制的值，然后选择 [复制] » [复制单元格]。

这样做不会复制列标题—只会将数据复制到剪贴板，且不会加引号。

现在便可将单元格内容粘贴到其它应用程序中。

在 Interactive SQL 结果集中对列进行排序

◆ 在结果集中对列进行排序

- 单击 [结果] 选项卡中的列标题，可按该列对结果进行排序。

[结果] 选项卡未包含整个结果集时，会提示您获取其余结果。否则，只会对当前已获取的结果进行排序。

打开多个窗口

可以打开多个 Interactive SQL 窗口。每个窗口对应于一个单独的数据库连接。您可以同时连接到不同数据库服务器上的两个（或多个）不同的数据库，也可以打开到单个数据库的并发连接。

◆ 打开新的 Interactive SQL 窗口

1. 从 [窗口] 菜单中，选择 [新建窗口]。

提示

如果设置了 SQLCONNECT 环境变量，或者已连接到某个 SQL Anywhere 数据库，则数据库服务器在提示您输入信息之前会先尝试使用此信息连接到数据库。同样，如果设置了 ULCONNECT 环境变量，或者已连接到某个 UltraLite 数据库，则数据库服务器在提示您输入信息之前会先尝试使用此信息连接到数据库。如果上述尝试失败，或者尚未连接到数据库，则会出现 [连接] 窗口。

2. 在 [连接] 窗口中，输入数据库的连接信息，然后单击 [确定] 进行连接。

在 Interactive SQL 的标题栏中即会出现连接信息（包括数据库名、用户 ID 和数据库服务器名）。

使用 [SQL] 菜单中的 [连接] 和 [断开连接] 项或通过执行 CONNECT 或 DISCONNECT 语句，也可以连接到数据库或与数据库断开连接。

使用源代码控制集成

Interactive SQL 与第三方源代码控制系统集成在一起，从而使您可以从 Interactive SQL 内对文件执行许多常见的源代码控制操作。在 Windows 上，Interactive SQL 可以集成大多数支持 Microsoft 公用源代码控制 API (SCC) 的源代码控制产品，包括 Microsoft Visual SourceSafe。要在 Windows 和其它操作系统上使用不支持 SCC API 的源代码控制产品，请指定对应每个源代码控制操作的命令行。这些命令的输出出现在日志窗口中。

Interactive SQL 支持以下任务（只要该任务在源代码控制产品中受支持）：

- 打开源代码控制项目
- 获取
- 签入
- 签出
- 撤消签出
- 比较版本
- 显示文件历史记录
- 显示文件属性
- 运行源代码控制管理器

如果底层源代码控制程序不支持某项操作，则其相应的菜单项将被禁用。例如，Visual SourceSafe 支持上述所有操作，但使用自定义（命令行）源代码控制系统时不支持打开源代码控制项目或运行源代码控制管理器。

有关所支持的操作的详细信息，请参见：

- “从 Interactive SQL 打开源代码控制项目” 一节第 660 页
- “从 Interactive SQL 签出文件” 一节第 661 页
- “从 Interactive SQL 签入文件” 一节第 662 页
- “其它源代码控制操作” 一节第 662 页

在尝试从 Interactive SQL 使用源代码控制程序之前，应先熟悉其操作。

将 Interactive SQL 配置为使用源代码控制

必须先将 Interactive SQL 配置为使用源代码控制，才能对文件执行源代码控制操作，如签入和签出文件、比较文件的不同版本以及查看文件的历史记录。

如果是在 Windows 计算机上运行 Interactive SQL 且该计算机装有支持 Microsoft SCC API 的源代码控制产品，则可以使用该产品或使用自定义（面向命令行的）系统。

配置 SCC 源代码控制系统

◆ 在带有 SCC 的 Windows 上配置 Interactive SQL 源代码控制

1. 单击 [工具] » [选项]。
2. 在左窗格中，单击 [源代码控制]。
3. 单击 [启用源代码控制集成]。
4. 单击 [确定]。

配置其它源代码控制系统

◆ 使用命令行界面配置 Interactive SQL 源代码控制系统

1. 单击 [工具] » [选项]。
2. 在左窗格中，单击 [源代码控制]。
3. 单击 [启用源代码控制集成]。
4. 单击 [配置]。
5. 在 [自定义源代码控制选项] 窗口中，单击 [重置]。
6. 从列表中选择源代码控制系统，然后单击 [确定]。
7. 根据需要编辑列表中的命令，方法是从 [源代码控制操作] 列表中选择某项操作，然后在 [命令行] 窗格中键入相应的命令。

在 [源代码控制操作] 列表中为系统定义命令时，使用占位符 [FILENAME] 来表示运行命令时所用文件的名称。例如，在 Perforce 中用于提交文件的命令是 `p4 submit [FILENAME]`。此列表中以粗体显示的操作已定义了命令，而以普通字体显示的操作则没有定义命令。



如果没有为操作指定命令行，则会禁用 [文件] » [源代码控制] 菜单中的菜单项。

提示

可以将源代码控制命令行导出到外部文件，方法是在 [自定义源代码控制选项] 窗口中单击 [导出]（选择 [工具] » [选项]，然后在 [源代码控制] 窗格上单击 [配置]，即可访问该窗口）。之后可以通过在此窗口中单击 [导入] 来重新读入这些命令。如果需要在多台计算机上配置 Interactive SQL 源代码控制命令行，这样做可能会很有用。

- 单击 [确定]，然后再次单击 [确定]。

从 Interactive SQL 打开源代码控制项目

某些源代码控制产品要求先打开源代码控制项目，才能执行任何其它的源代码控制操作。项目的确切定义取决于所使用的源代码控制系统。通常，它是指处于源代码控制下的一组文件，以及本地文件系统中放置文件工作副本的位置。通常必须向源代码控制系统提供某些凭证（如用户 ID 和口令）方可打开项目。

如果源代码控制系统支持打开源代码控制项目，则会启用 [文件] » [源代码控制] » [打开源代码控制项目] 菜单项。从 [文件] 菜单选择此选项时，会打开源代码控制特有的一个用于打开项目的窗口。

一旦打开项目，就不必再次打开它了，甚至在后续的 Interactive SQL 会话中也是如此。该项目将会自动为您打开。

从 Interactive SQL 签出文件

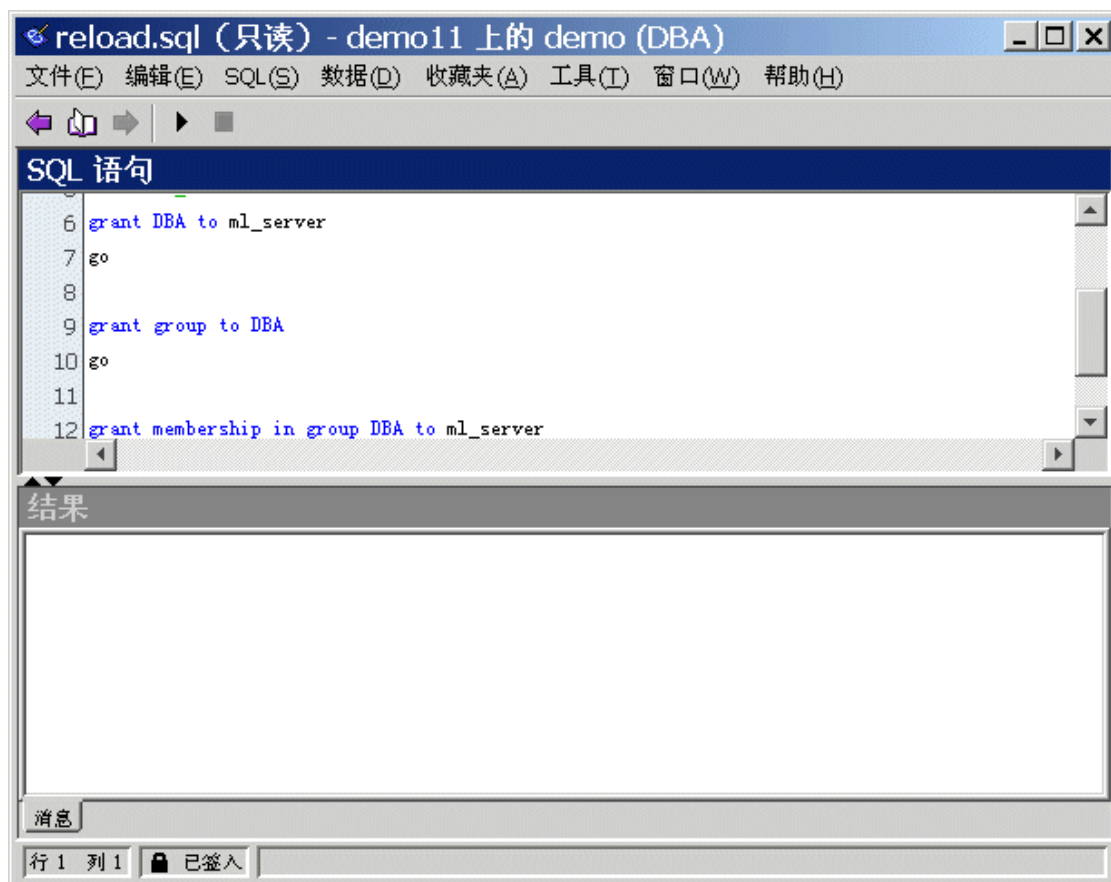
在 Interactive SQL 中打开文件后，便可采用以下两种方法签出该文件：在 [SQL 语句] 窗格中修改其内容，或使用 [文件] 菜单上的命令。

如果您在为 Interactive SQL 配置源代码控制选项时选择了 [修改编辑器内容时自动签出文件]，则当您在 [SQL 语句] 窗格中修改文件内容时，Interactive SQL 会尝试签出该文件。

◆ 使用 Interactive SQL 的 [文件] 菜单签出文件

1. 选择 [文件] » [打开]，然后浏览到所要打开的文件。

在 [Interactive SQL] 窗口底部的状态栏上即会出现文件状态。状态为 [已签入]、[已签出] 或 [Not Controlled] 三者之一。已签入的文件被假定为只读的，并且在 Interactive SQL 标题栏中会出现 [只读] 字样。以下示例中的文件已签入：



2. 选择 [文件] » [源代码控制] » [签出] 签出文件。

在签出过程中可能会提示您输入注释或其它选项，这取决于所使用的源代码控制产品。

小心

如果使用的是 SCC 兼容的源代码控制系统，该状态始终都是准确的。但是，如果使用自定义源代码控制系统，则状态取决于文件是否为只读。只读文件被假定为已签入，但不会对可编辑文件做任何假定，因为它们既可能是已签出的又可能是不受控制的。

从 Interactive SQL 签入文件

完成对文件所做的编辑后，可从 Interactive SQL 将其签入回来。

◆ 从 Interactive SQL 签入文件

1. 选择 [文件] » [源代码控制] » [签入]。
2. 收到提示时输入签入注释。

其它源代码控制操作

除了打开源代码控制项目以及签入和签出文件之外，Interactive SQL 还支持其它若干源代码控制操作。这些操作的可用性取决于所使用的源代码控制系统。这些操作可通过 Interactive SQL 中的 [文件] » [源代码控制] 菜单进行访问。

- **获取** 此操作会获取 [SQL 语句] 窗格中当前已打开文件的最新副本。
- **撤消签出** 如果在签出了文件后想要放弃所做的更改，请选择 [文件] » [源代码控制] » [撤消签出]。这样便会废弃文件的工作副本，然后下载该文件在源代码控制档案中的副本。
- **比较版本** 此操作会将已打开文件的工作副本与源代码控制档案中的版本进行比较。
- **历史记录** 此操作会显示对已打开的文件所执行的源代码控制操作（通常是签入）列表。
- **属性** 此操作会显示与已打开的文件相关联的源代码控制属性列表。
- **运行源代码控制管理器** 此操作会启动源代码控制系统的管理程序。例如，如果使用的是 Microsoft Visual SourceSafe，便会启动 Visual SourceSafe Explorer。

Interactive SQL SQL 语句

下表提供指向 SQL Anywhere 文档中有关 Interactive SQL 用户可以使用的 SQL 语句的链接。

“CLEAR 语句 [Interactive SQL]” 一节 《SQL Anywhere 服务器 - SQL 参考》

“CONFIGURE 语句 [Interactive SQL]” 一节 《SQL Anywhere 服务器 - SQL 参考》

“CONNECT 语句 [ESQL] [Interactive SQL]” 一节 《SQL Anywhere 服务器 - SQL 参考》

- “DESCRIBE 语句 [Interactive SQL]” 一节 《SQL Anywhere 服务器 - SQL 参考》
- “DISCONNECT 语句 [ESQL] [Interactive SQL]” 一节 《SQL Anywhere 服务器 - SQL 参考》
- “EXIT 语句 [Interactive SQL]” 一节 《SQL Anywhere 服务器 - SQL 参考》
- “HELP 语句 [Interactive SQL]” 一节 《SQL Anywhere 服务器 - SQL 参考》
- “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》
- “PARAMETERS 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》
- “SET CONNECTION 语句 [Interactive SQL] [ESQL]” 一节 《SQL Anywhere 服务器 - SQL 参考》
- “SET OPTION 语句 [Interactive SQL]” 一节 《SQL Anywhere 服务器 - SQL 参考》
- “START ENGINE 语句 [Interactive SQL]” 一节 《SQL Anywhere 服务器 - SQL 参考》
- “START LOGGING 语句 [Interactive SQL]” 一节 《SQL Anywhere 服务器 - SQL 参考》
- “STOP LOGGING 语句 [Interactive SQL]” 一节 《SQL Anywhere 服务器 - SQL 参考》
- “SYSTEM 语句 [Interactive SQL]” 一节 《SQL Anywhere 服务器 - SQL 参考》

另请参见

- “使用 SQL 语句参考” 一节 《SQL Anywhere 服务器 - SQL 参考》

Interactive SQL 键盘快捷方式

Interactive SQL 提供以下键盘快捷方式：

按键	说明
Alt+F4	关闭 Interactive SQL。
Alt+左箭头键	显示历史列表中的上一个 SQL 语句。
Alt+右箭头键	显示历史列表中的下一个 SQL 语句。
Ctrl+Backspace	删除光标左边的单词。
Ctrl+Break	中断正在执行的 SQL 语句。
Ctrl+A	选择活动窗格中的所有文本。 在 [结果] 窗格中，如果结果不是整个结果集，则会提示您获取其余行。否则，只会选择当前已获取的结果。请参见“从 Interactive SQL 结果集中复制行”一节第 656 页。

按键	说明
Ctrl+C	在 [结果] 窗格中，将选定的行和列标题复制到剪贴板。 在 [SQL 语句] 窗格中，将选定的文本复制到剪贴板。
Ctrl+Del	删除光标右边的单词。
Ctrl+End	将光标移动到当前窗格的底部。
Ctrl+F	打开 [查找/替换] 窗口。
Ctrl+G	将光标移到 [SQL 语句] 窗格中的特定行。
Ctrl+H	显示所执行 SQL 语句的历史记录。
Ctrl+Home	将光标移动到当前窗格的顶部。
Ctrl+L	从 [SQL 语句] 窗格中删除当前行并将该行放入剪贴板中。
Ctrl+Shift+L	删除当前行。
Ctrl+N	清除 Interactive SQL 窗口的内容并关闭当前文件（如果有）。
Ctrl+O	打开文件。
Ctrl+P	打印 [SQL 语句] 窗格中的内容；等同于从 [工具] 菜单选择 [选项]，然后选择 [编辑器]，随后单击 [打印] 选项卡。
Ctrl+Q	打开查询编辑器。 查询编辑器可帮助您建立 SQL 查询。建完查询后，单击 [确定] 可将其导出到 [SQL 语句] 窗格中。
Ctrl+S	将 [SQL 语句] 窗格的内容保存到指定文件。
Ctrl+U	将选定内容更改为小写字符。
Ctrl+Shift+U	将选定内容更改为大写字符。
Ctrl+V	粘贴选定的文本。
Ctrl+X	剪切选定的文本。
Ctrl+Y	重复上一个操作。
Ctrl+Z	撤消上一个操作。
Ctrl+]	将光标移动到匹配的括号。使用此快捷方式可匹配小括号、大括号、中括号和尖括号。

按键	说明
Ctrl+Shift+]]	将选择范围扩展到匹配的括号。使用此快捷方式可匹配小括号、大括号、中括号和尖括号。
Ctrl+减号 (-)	<p>添加或删除双连字符 (--) SQL 注释指示符。</p> <p>要将现有文本变成注释，请在 [SQL 语句] 窗格中选择文本，然后按 Ctrl+减号。SQL 注释指示符会添加到选定文本各行的开头。</p> <p>如果未选定任何文本，注释指示符会添加到当前行的开头。</p> <p>要删除注释指示符，请选择文本然后按 Ctrl+减号。</p> <p>请参见“注释”一节《SQL Anywhere 服务器 - SQL 参考》。</p>
Ctrl+正斜线 (/)	<p>添加或删除双斜线 (//) SQL 注释指示符。</p> <p>要将现有文本变成注释，请在 [SQL 语句] 窗格中选择文本，然后按 Ctrl+正斜线。SQL 注释指示符会添加到选定文本各行的开头。</p> <p>如果未选定任何文本，注释指示符会添加到当前行的开头。</p> <p>要删除注释指示符，请选择文本然后按 Ctrl+正斜线。</p> <p>请参见“注释”一节《SQL Anywhere 服务器 - SQL 参考》。</p>
Ctrl+上箭头键	选择 [SQL 语句] 窗格中光标所在语句之前的 SQL 语句。
Ctrl+下箭头键	选择 [SQL 语句] 窗格中光标所在语句之后的 SQL 语句。
Ctrl+句点 (.)	选择 [SQL 语句] 窗格中光标所在位置的整个 SQL 语句。
Ctrl+逗号 (,)	选择 [SQL 语句] 窗格中光标所在的行。
Ctrl+Shift+句点 (.)	<p>增加 [SQL 语句] 窗格中选定文本的行缩进。</p> <p>如果未选定任何文本，则会将缩进应用到当前行。</p> <p>可以更改缩进的空格数；方法是从 [工具] 菜单选择 [选项]，然后选择 [编辑器]，随后单击 [制表符] 选项卡。</p>
Ctrl+Shift+逗号 (,)	<p>减少 [SQL 语句] 窗格中选定文本的行缩进。</p> <p>如果未选定任何文本，则会将缩进应用到当前行。</p> <p>可以更改缩进的空格数；方法是从 [工具] 菜单选择 [选项]，然后选择 [编辑器]，随后单击 [制表符] 选项卡。</p>
Esc	清除 [SQL 语句] 窗格。
F1	打开联机帮助。
F2	编辑结果集中选定的值。可在行内使用 Tab 键在各列之间切换。

按键	说明
F3	查找指定文本的下一个出现位置。
Shift+F3	查找选定文本的上一个出现位置。
F5	执行 [SQL 语句] 窗格中的所有文本。 也可通过单击工具栏上的 [执行所有 SQL 语句] 或选择 [SQL] » [执行] 来执行此操作。
Shift+F5	针对 [SQL 语句] 窗格中的指定语句打开计划查看器。指定语句不会执行；要在计划查看器中执行语句，请单击 [获取计划]。
F7	显示 [查寻表名] 窗口。 在此窗口中，可以查找并选择一个表，然后按 Enter 键将表名插入到 [SQL 语句] 窗格中光标所在的位置。或者，在列表选定某个表后，再次按 F7 键显示该表中的列。然后，可以选择某一列并按 Enter 键将列名插入 [SQL 语句] 窗格中光标所在的位置。
F8	显示 [查寻过程名] 窗口。 在此窗口中，可以查找并选择一个过程，然后按 Enter 键将过程名插入到 [SQL 语句] 窗格中光标所在的位置。
F9	执行 [SQL 语句] 窗格中选定的文本。 如果未选定任何文本，则会执行所有的语句。 也可通过单击工具栏上的 [执行所选语句] 或选择 [SQL] » [执行] 来执行此操作。
Shift+F9	执行选定的 SQL 语句，然后选中下一条语句。此快捷方式用于逐步执行一系列 SQL 语句。请参见“从 Interactive SQL 执行 SQL 语句”一节第 641 页。
Shift+F10	显示焦点所在区域的快捷菜单。 此键盘快捷方式与右击某个区域的作用相同。
F11	如果 Interactive SQL 未连接到数据库，则打开 [连接] 窗口。
F12	断开 Interactive SQL 与当前数据库的连接。
Home	将光标移动到当前行的开始位置或当前行的第一个单词。
Shift+Home	将选择范围扩展到当前行上文本的开始位置。
Page Down	在当前窗格中向下移动一页。

按键	说明
Page Up	在当前窗格中向上移动一页。

Interactive SQL 选项

SET OPTION 语句用于更改以下 Interactive SQL 选项的值。请参见“SET OPTION 语句 [Interactive SQL]”一节《SQL Anywhere 服务器 - SQL 参考》。

选项	值	缺省
“auto_commit 选项 [Interactive SQL]”一节第 668 页	On、Off	Off
“auto_refetch 选项 [Interactive SQL]”一节第 669 页	On、Off	On
“bell 选项 [Interactive SQL]”一节第 669 页	On、Off	On
“command_delimiter 选项 [Interactive SQL]”一节第 670 页	字符串	' ; '
“commit_on_exit 选项 [Interactive SQL]”一节第 671 页	On、Off	On
“default_isql_encoding 选项 [Interactive SQL]”一节第 671 页	字符串	空字符串
“echo 选项 [Interactive SQL]”一节第 672 页	On、Off	On
“input_format 选项 [Interactive SQL]”一节第 672 页	TEXT、FIXED	TEXT
“isql_allow_read_client_file 选项 [Interactive SQL]”一节第 673 页	On、Off、Prompt	Prompt
“isql_allow_write_client_file 选项 [Interactive SQL]”一节第 674 页	On、Off、Prompt	Prompt
“isql_command_timing 选项 [Interactive SQL]”一节第 674 页	On、Off	On
“isql_escape_character 选项 [Interactive SQL]”一节第 675 页	字符	'\ '

选项	值	缺省
“isql_field_separator 选项 [Interactive SQL]” 一节第 676 页	字符串	' , '
“isql_maximum_displayed_rows 选项 [Interactive SQL]” 一节第 677 页	All 或一个非负整数	500
“isql_print_result_set 选项 [Interactive SQL]” 一节第 677 页	Last、All、None	Last
“isql_quote 选项 [Interactive SQL]” 一节第 678 页	字符串	'
“isql_show_multiple_result_sets [Interactive SQL]” 一节第 678 页	On、Off	Off
“nulls 选项 [Interactive SQL]” 一节第 679 页	字符串	'(NULL)'
“on_error 选项 [Interactive SQL]” 一节第 679 页	Stop、Continue、Prompt、Exit、Notify_Continue、Notify_Stop、Notify_Exit	Prompt
“output_format 选项 [Interactive SQL]” 一节第 680 页	TEXT、FIXED、HTML、SQL、XML	TEXT
“output_length 选项 [Interactive SQL]” 一节第 681 页	整数	0
“output_nulls 选项 [Interactive SQL]” 一节第 681 页	字符串	空字符串
“truncation_length 选项 [Interactive SQL]” 一节第 682 页	整数	256

auto_commit 选项 [Interactive SQL]

控制是否在各条语句后执行 COMMIT。

允许值

On、Off

缺省值

Off

注释

如果 auto_commit 为 ON，则会在每条语句成功完成后执行数据库 COMMIT。

缺省情况下，只有在用户发出 COMMIT 或 ROLLBACK 语句或导致自动提交的 SQL 语句（例如 CREATE TABLE 语句）时才执行 COMMIT 或 ROLLBACK 语句。

在 Interactive SQL 中使用数据源

缺省情况下，ODBC 在自动提交模式下工作。即使已在 Interactive SQL 中将 auto_commit 选项设置为 Off，ODBC 的设置也将替换 Interactive SQL 的设置。可以使用 SQL_ATTR_AUTOCOMMIT 连接属性更改 ODBC 的设置。ODBC 自动提交与 chained 选项无关。

另请参见

- “commit_on_exit 选项 [Interactive SQL]” 一节第 671 页
- “SET OPTION 语句 [Interactive SQL]” 一节 《SQL Anywhere 服务器 - SQL 参考》

auto_refetch 选项 [Interactive SQL]

控制在删除、更新和插入之后是否再次读取查询结果。

允许值

On、Off

缺省值

On

注释

如果 auto_refetch 为 On，则在执行任意 INSERT、UPDATE 或 DELETE 语句后，会从数据库中重新读取在 Interactive SQL 的 [结果] 窗格的 [结果] 选项卡中显示的当前查询结果。根据查询的复杂程度，这可能会花费一段时间。因此，可以将其关闭。

另请参见

- “SET OPTION 语句 [Interactive SQL]” 一节 《SQL Anywhere 服务器 - SQL 参考》

bell 选项 [Interactive SQL]

控制出现错误时是否发出铃声。

允许值

On、Off

缺省值

On

注释

根据您的喜好设置此选项。

另请参见

- “SET OPTION 语句 [Interactive SQL]” 一节 《SQL Anywhere 服务器 - SQL 参考》

command_delimiter 选项 [Interactive SQL]

设置在 Interactive SQL 中指示语句结束的字符串。

允许值

字符串

缺省值

分号 (;)

注释

通常不需要更改命令分隔符。应将其保留为分号。

如果不用分号或其它字符串作为语句分隔符，也可以在一行的开头单独输入分隔符 **go**。请参见“批处理简介”一节 《SQL Anywhere 服务器 - SQL 的用法》。

在行开头自成一行指定 **go** 时，始终会将其识别为命令分隔符，即使将 `command_delimiter` 选项设置为其它值也是如此。

`command_delimiter` 值可以是具有以下限制的任意字符串：

- 如果分隔符包含 &（和号）、*（星号）、@（at 符号）、:（冒号）、.（句号）、=（等号）、（（左括号）、）（右括号）或|（竖线）中的任何一个，则分隔符不能包含其它任何字符。例如，* 是一个有效分隔符，但 ** 却不是。
- 不应将现有的关键字用作命令分隔符。请参见“关键字”一节 《SQL Anywhere 服务器 - SQL 参考》。
- 命令分隔符可以是任意的字符序列（可包括数字、字母和标点符号），但它不能包含嵌入空格。另外，它可以包含分号，但分号只能作为第一个字符。

如果将命令分隔符设置为一个以标识符中的有效字符开头的字符串，则该命令分隔符必须跟在一个空格的后面。命令分隔符区分大小写。新命令分隔符必须用单引号括起来。如果命令分隔符是一个分号（缺省值），则不需要在分号前加空格。

另请参见

- “Interactive SQL 实用程序 (dbisql)” 一节第 740 页
- “SET OPTION 语句 [Interactive SQL]” 一节 《SQL Anywhere 服务器 - SQL 参考》

示例

以下示例将命令分隔符设置为代字号：


```
SET OPTION command_delimiter='~';  
MESSAGE 'hello'~
```

还可以使用 Interactive SQL `-d` 选项设置命令分隔符，而不用在 `.sql` 文件中加入 `SET OPTION command_delimiter` 语句。例如，如果有一个名为 `test.sql` 的脚本文件使用代字号 (`~`) 作为命令分隔符，则可以运行：

```
dbisql -d "~" test.sql
```

commit_on_exit 选项 [Interactive SQL]

控制 Interactive SQL 断开连接或关闭时的行为。

允许值

On、Off

Default

On

注释

控制在离开 Interactive SQL 时是否进行 COMMIT 或 ROLLBACK 操作。当 `commit_on_exit` 设置为 On 时，将会进行 COMMIT 操作。

另请参见

- [“SET OPTION 语句 \[Interactive SQL\]”一节 《SQL Anywhere 服务器 - SQL 参考》](#)

default_isql_encoding 选项 [Interactive SQL]

指定 READ、INPUT 和 OUTPUT 语句应使用的代码页。

允许值

标识符或字符串

缺省值

使用系统代码页（空字符串）

作用域

只能设置为临时选项，保持到当前连接结束。

注释

此选项用于指定读取或写入文件时使用的代码页。无法对其进行永久性设置。缺省代码页是运行时所在平台的缺省代码页。在英文版 Windows 计算机上，缺省代码页为 1252。

Interactive SQL 会按如下方式确定用于特定 INPUT、OUTPUT 或 READ 语句的代码页，其中，列表中先出现的代码页值优先于后出现的值：

- 在 INPUT、OUTPUT 或 READ 语句的 ENCODING 子句中指定的代码页
- 用 default_isql_encoding 选项指定的代码页（如果设置了此选项）
- 运行 Interactive SQL 的计算机的缺省代码页

有关代码页和字符集的详细信息，请参见“有关国际语言和字符集的任务”一节第 399 页。

另请参见

- “READ 语句 [Interactive SQL]”一节 《SQL Anywhere 服务器 - SQL 参考》
- “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》
- “字符集、编码和归类概述”一节第 384 页
- “SET OPTION 语句 [Interactive SQL]”一节 《SQL Anywhere 服务器 - SQL 参考》

示例

将编码设置为 UTF-16（用于读取 Unicode 文件）：

```
SET TEMPORARY OPTION default_isql_encoding = 'UTF-16';
```

echo 选项 [Interactive SQL]

控制语句在执行前是否回写到日志文件。

允许值

On、Off

Default

On

注释

在使用 READ 语句执行 Interactive SQL 命令文件，或通过选择 [文件] » [运行脚本] 在 Interactive SQL 中运行命令文件时，此选项最为有用。必须开启记录功能方可使此选项生效。请参见“START LOGGING 语句 [Interactive SQL]”一节 《SQL Anywhere 服务器 - SQL 参考》。

另请参见

- “SET OPTION 语句 [Interactive SQL]”一节 《SQL Anywhere 服务器 - SQL 参考》

input_format 选项 [Interactive SQL]

设置 INPUT 语句所期望的缺省数据格式。

允许值

字符串（有关允许值，请参见以下内容）

缺省值

TEXT

注释

允许的输入格式为：

- **TEXT** 输入行被假定为文本字符，每行代表一数据行，值以逗号分隔。字母字符串可以用撇号（单引号）或引号（双引号）括起来。包含逗号的字符串必须用单引号或双引号括起来。如果使用了单引号或双引号，要将该引号字符双写才可在字符串内使用它。如果需要，也可以使用 **DELIMITED BY** 子句指定缺省值（逗号 (,)）以外的其它分隔符字符串。

另外，还识别其它三个特殊序列。两个字符组成的串 `\n` 表示换行符，`\\` 表示单个反斜线字符，序列 `\xDD`（其中的 *DD* 是字符的十六进制表示形式）表示十六进制码为 *DD* 的字符。

- **FIXED** 输入行采用固定长度格式。

另请参见

- “**INPUT** 语句 [Interactive SQL]” 一节 《SQL Anywhere 服务器 - SQL 参考》
- “**SET OPTION** 语句 [Interactive SQL]” 一节 《SQL Anywhere 服务器 - SQL 参考》

isql_allow_read_client_file 选项 [Interactive SQL]

控制是否允许此连接读取客户端文件。

允许值

On、Off、Prompt

Default

Prompt

注释

此选项控制数据库服务器是否可读取客户端计算机上的文件。On 表示允许读取。Off 表示不允许读取。Prompt 表示提示用户选择要采取的操作。

此选项基于每个连接存储，且仅在此连接持续期间存在。可以使用 **SET TEMPORARY OPTION** 语句设置此选项。如果遗漏了 **TEMPORARY** 关键字，Interactive SQL 会报错。

如果从存储过程或触发器中执行 **LOAD TABLE**，此选项允许无需用户干预即读取数据文件。

要读取客户端计算机上的文件，需要具有 **READCLIENTFILE** 权限。

另请参见

- “访问客户端计算机上的数据”一节 《SQL Anywhere 服务器 - SQL 的用法》
- “READCLIENTFILE 特权”一节第 426 页
- “READ_CLIENT_FILE 函数 [String]”一节 《SQL Anywhere 服务器 - SQL 参考》
- “SET OPTION 语句 [Interactive SQL]”一节 《SQL Anywhere 服务器 - SQL 参考》
- “LOAD TABLE 语句”一节 《SQL Anywhere 服务器 - SQL 参考》
- “allow_read_client_file 选项 [数据库]”一节第 478 页
- “allow_write_client_file 选项 [数据库]”一节第 479 页
- “isql_allow_write_client_file 选项 [Interactive SQL]”一节第 674 页
- “客户端数据安全性”一节 《SQL Anywhere 服务器 - SQL 的用法》

isql_allow_write_client_file 选项 [Interactive SQL]

控制是否允许此连接写入客户端文件。

允许值

On、Off、Prompt

Default

Prompt

注释

此选项控制数据库服务器是否可写入客户端计算机上的文件。On 表示允许写入。Off 表示不允许写入。Prompt 表示提示用户选择要采取的操作。

此选项基于每个连接存储，且仅在此连接持续期间存在。可以使用 SET TEMPORARY OPTION 语句设置此选项。如果遗漏了 TEMPORARY 关键字，Interactive SQL 会报错。

要写入客户端计算机上的文件，需要具有 WRITECLIENTFILE 权限。

另请参见

- “访问客户端计算机上的数据”一节 《SQL Anywhere 服务器 - SQL 的用法》
- “WRITECLIENTFILE 特权”一节第 427 页
- “WRITE_CLIENT_FILE 函数 [String]”一节 《SQL Anywhere 服务器 - SQL 参考》
- “SET OPTION 语句 [Interactive SQL]”一节 《SQL Anywhere 服务器 - SQL 参考》
- “UNLOAD 语句”一节 《SQL Anywhere 服务器 - SQL 参考》
- “allow_write_client_file 选项 [数据库]”一节第 479 页
- “allow_read_client_file 选项 [数据库]”一节第 478 页
- “isql_allow_read_client_file 选项 [Interactive SQL]”一节第 673 页
- “客户端数据安全性”一节 《SQL Anywhere 服务器 - SQL 的用法》

isql_command_timing 选项 [Interactive SQL]

控制是否对 SQL 语句计时。

允许值

On、Off

Default

On

注释

此布尔选项控制是否对 SQL 语句计时。如果将该选项设置为 **On**，则在执行一条语句后会在 [消息] 窗格中显示执行时间。如果将该选项设置为 **Off**，则不会显示时间。

也可以在 [选项] 窗口的 [消息] 选项卡中设置此选项。

另请参见

- “SET OPTION 语句 [Interactive SQL]” 一节 《SQL Anywhere 服务器 - SQL 参考》

isql_escape_character 选项 [Interactive SQL]

控制在导出到文本文件时用于转换数据内不可打印字符的转义字符。

允许值

任意单个字符

缺省值

反斜线 (\)

注释

当 Interactive SQL 导出含有不可打印字符（如回车符）的字符串时，它会将每个不可打印字符转换为十六进制格式，并在其前面加上一个转义字符。如果 OUTPUT 语句不包含 ESCAPE CHARACTER 子句，则会在输出中使用您为此设置所指定的字符。仅在导出到文本文件时才会用到此项设置。

另请参见

- “isql_quote 选项 [Interactive SQL]” 一节第 678 页
- “SET OPTION 语句 [Interactive SQL]” 一节 《SQL Anywhere 服务器 - SQL 参考》

示例

创建一个表，其中包含一个含有嵌入回车符（在 INSERT 语句中以 "\n" 表示）的字符串值。然后以 # 符号作为转义字符将数据导出到 *c:\escape.txt*。

```
CREATE TABLE escape_test( text varchar(10 ) );
INSERT INTO escape_test VALUES( 'one\ntwo' );
SET OPTION isql_escape_character='#';
SELECT * FROM escape_test;
OUTPUT TO c:\escape.txt FORMAT TEXT;
```

此代码会在 *escape.txt* 中置入以下数据：

```
'one#x0Atwo'
```

井号 (#) 是转义字符，**x0A** 是 **\n** 字符的十六进制等效值。

起始和结束字符（在本例中为单引号）取决于 `isql_quote` 设置。

isql_field_separator 选项 [Interactive SQL]

控制用于在导出到文本文件的数据中分隔值的缺省字符串。

允许值

字符串

缺省值

逗号 (,)

注释

控制用于在导出到文本文件的数据中分隔（或定界）值的缺省字符串。如果 `OUTPUT` 语句不包含 `DELIMITED BY` 子句，则会使用此项设置的值。

另请参见

- [“isql_quote 选项 \[Interactive SQL\]” 一节第 678 页](#)
- [“SET OPTION 语句 \[Interactive SQL\]” 一节 《SQL Anywhere 服务器 - SQL 参考》](#)
- [“isql_escape_character 选项 \[Interactive SQL\]” 一节第 675 页](#)

示例

第一个示例在导出到 `c:\Employees.txt` 的数据中将字段分隔符设置为冒号。

```
SET OPTION isql_field separator=':';
SELECT Surname, GivenName FROM Employees WHERE EmployeeID < 150;
OUTPUT TO c:\Employees.txt FORMAT TEXT;
```

此代码会在 `Employees.txt` 中置入以下数据：

```
'Whitney': 'Fran'
'Cobb': 'Matthew'
'Chin': 'Philip'
'Jordan': 'Julie'
```

起始和结束字符（在本例中为单引号）取决于 `isql_quote` 设置。

下一个示例在导出到 `c:\Employees.txt` 的数据中将字段分隔符设置为制表符。

```
SET OPTION isql_field separator='\t';
SELECT Surname, GivenName FROM Employees WHERE EmployeeID < 150;
OUTPUT TO c:\Employees.txt FORMAT TEXT;
```

此代码会在 `Employees.txt` 中置入以下数据：

```
Surname GivenName
'Whitney' 'Fran'
'Cobb' 'Matthew'
```

```
'Chin' 'Philip'  
'Jordan' 'Julie'
```

起始和结束字符（在本例中为单引号）取决于 `isql_quote` 设置。转义字符（在本例中为反斜线）取决于 `isql_escape_character` 设置。

isql_maximum_displayed_rows 选项 [Interactive SQL]

指定可在 Interactive SQL 的 [结果] 窗格中显示的最大行数。

允许值

ALL 或一个非负整数

缺省值

500

注释

使用此选项可以指定显示在 [结果] 窗格中的最大行数。也可以在 Interactive SQL 的 [选项] 窗口中设置此选项的值。

小心

显示大型结果集时，Interactive SQL 可能发生内存不足。如果出现此问题，Interactive SQL 将报告该问题，且不显示结果集。

另请参见

- “SET OPTION 语句 [Interactive SQL]” 一节 《SQL Anywhere 服务器 - SQL 参考》

isql_print_result_set 选项 [Interactive SQL]

指定运行 `.sql` 文件时打印的结果集。

允许值

LAST、ALL、NONE

缺省值

LAST

注释

仅当将 Interactive SQL 作为命令程序运行时（例如，运行 `.sql` 文件时），`isql_print_result_set` 选项才会生效。

使用此选项可以指定在运行 `.sql` 文件时打印的结果集。

可以选择以下打印选项之一：

- **LAST** 打印文件的最后一条语句的结果集。
- **ALL** 打印文件中返回结果集的每条语句的结果集。
- **NONE** 不打印任何结果集。

另请参见

- [“SET OPTION 语句 \[Interactive SQL\]”一节 《SQL Anywhere 服务器 - SQL 参考》](#)

isql_quote 选项 [Interactive SQL]

控制导出到文本文件的数据中所有字符串缺省的开始和结束字符串。

允许值

字符串

缺省值

单个撇号 (')

注释

控制导出到文本文件的数据中所有字符串缺省的开始和结束字符串。如果 OUTPUT 语句不包含 QUOTE 子句，则在缺省情况下使用此值。

另请参见

- [“isql_field_separator 选项 \[Interactive SQL\]”一节第 676 页](#)
- [“SET OPTION 语句 \[Interactive SQL\]”一节 《SQL Anywhere 服务器 - SQL 参考》](#)

示例

将所有字符串缺省的开始和结束字符串更改为双引号字符。

```
SET OPTION isql_quote='";
SELECT Surname, GivenName FROM Employees WHERE EmployeeID < 150;
OUTPUT TO c:\Employees.txt FORMAT TEXT;
```

此代码会在 *Employees.txt* 中置入以下数据：

```
"Whitney", "Fran"
"Cobb", "Matthew"
"Chin", "Philip"
"Jordan", "Julie"
```

分隔符（在本例中为逗号）取决于 isql_field_separator 设置。

isql_show_multiple_result_sets [Interactive SQL]

指定在 Interactive SQL 的 [结果] 窗格中能否出现多个结果集。

允许值

On、Off

缺省值

Off

注释

如果在执行返回多个 SELECT 语句的过程时想让 Interactive SQL 在 [结果] 窗格中显示多个结果集，请将此选项设置为 On。

每个结果集在 [结果] 窗格中均出现在单独的选项卡上。缺省情况下，Interactive SQL 不显示多个结果集。此选项的设置同样适用于作为命令程序运行的 Interactive SQL。

另请参见

- [“SET OPTION 语句 \[Interactive SQL\]”](#) 一节 《SQL Anywhere 服务器 - SQL 参考》

nulls 选项 [Interactive SQL]

指定在 Interactive SQL 中显示结果时如何显示数据库中的空值。

允许值

字符串

缺省值

(NULL)

注释

根据您的喜好设置此选项。请注意，在将结果集保存到文件时不会使用此值。保存到文件时所用的值由 output_nulls 选项指定。

另请参见

- [“SET OPTION 语句 \[Interactive SQL\]”](#) 一节 《SQL Anywhere 服务器 - SQL 参考》
- [“output_nulls 选项 \[Interactive SQL\]”](#) 一节第 681 页

on_error 选项 [Interactive SQL]

控制在 Interactive SQL 中执行语句期间遇到错误时所采取的操作。

允许值

字符串（有关允许值，请参见以下内容）

Default

Prompt

注释

控制在执行语句期间遇到错误时所采取的操作，如下所示：

- **Stop** Interactive SQL 停止执行语句。
- **Prompt** Interactive SQL 提示用户决定是否要继续。
- **Continue** 忽略错误，Interactive SQL 继续执行语句。
- **Exit** Interactive SQL 关闭。
- **Notify_Continue** 报告错误，并提示用户按 Enter 键或单击 [确定] 继续。
- **Notify_Stop** 报告错误，并提示用户按 Enter 键或单击 [确定] 停止执行语句。
- **Notify_Exit** 报告错误，并提示用户按 Enter 键或单击 [确定] 关闭 Interactive SQL。

执行 `.sql` 文件时，Stop 和 Exit 值是等效的。如果指定了这两个值中的任意一个，Interactive SQL 将会关闭。

另请参见

- “[SET OPTION 语句 \[Interactive SQL\]](#)” 一节 《[SQL Anywhere 服务器 - SQL 参考](#)》

output_format 选项 [Interactive SQL]

功能

为由 SELECT 语句检索到的被重定向到文件的数据或使用 OUTPUT 语句输出的数据设置缺省输出格式。

允许值

字符串（有关允许值，请参见以下内容）

Default

TEXT

注释

有效的输出格式为：

- **TEXT** 输出 TEXT 格式的文件，且文件中每行代表一数据行。所有的值都用逗号分隔，并且字符串括在撇号（单引号）中。可以使用 DELIMITED BY 和 QUOTE 子句更改分隔符和引号字符串。如果在 QUOTE 子句中指定 All，则所有的值（不仅是字符串）都将用引号引起来。
还可以使用其它三种特殊序列。两个字符组成的串 `\n` 表示换行符；`\\` 表示单个反斜线字符，序列 `\xDD` 表示十六进制码为 DD 的字符。
- **FIXED** 输出每列都有固定宽度的固定格式。可以使用 COLUMN WIDTH 子句指定每列的宽度。如果省略此子句，则每列的宽度均由该列的数据类型计算得出，并且大小足以存放该数据类型的任意值。在该格式中不输出列标题。

- **HTML** 输出格式为 HTML。
- **SQL** 输出结果是重新创建表中的信息所需的 Interactive SQL INPUT 语句。
- **XML** 输出结果是以 UTF-8 编码且包含嵌入式 DTD 的 XML 文件。二进制值在 CDATA 块中进行编码，块中的二进制数据呈现为两位十六进制数的字符串。

另请参见

- [“SET OPTION 语句 \[Interactive SQL\]”](#) 一节 《SQL Anywhere 服务器 - SQL 参考》

output_length 选项 [Interactive SQL]

控制 Interactive SQL 在将信息导出至外部文件时的列值长度。

允许值

非负整数

缺省值

0（没有截断）

注释

此选项控制 Interactive SQL 将数据导出到外部文件（一同使用输出重定向和 OUTPUT 语句）时的最大列值长度。此选项只对 TEXT、HTML 和 SQL 输出格式有影响。

另请参见

- [“SET OPTION 语句 \[Interactive SQL\]”](#) 一节 《SQL Anywhere 服务器 - SQL 参考》

output_nulls 选项 [Interactive SQL]

控制空值的导出方式。

允许值

字符串

缺省值

空字符串

注释

此选项控制 OUTPUT 语句写入空值的方式。每次在结果集中找到空值时，均会改为返回来自此选项的字符串。此选项只对 TEXT、HTML、FIXED 和 SQL 输出格式有影响。

另请参见

- [“SET OPTION 语句 \[Interactive SQL\]”](#) 一节 《SQL Anywhere 服务器 - SQL 参考》

truncation_length 选项 [Interactive SQL]

控制宽列的截断以使其适合在屏幕上显示。

允许值

整数

缺省值

256

注释

truncation_length 选项用于限制所显示列值的长度。单位是字符。值为 0 意味着不对列值进行截断。缺省截断长度为 256。

另请参见

- “SET OPTION 语句 [Interactive SQL]” 一节 《SQL Anywhere 服务器 - SQL 参考》

使用文本完成

Interactive SQL 和 Sybase Central 提供了文本完成选项，该选项可为您提供对象名称。可以配置文本完成，以完成以下任意或所有对象类型的名称：表、视图、列、存储过程和系统函数。

对于 SELECT、INSERT、UPDATE、DELETE 和 DESCRIBE 语句，所列出的可能建议与您在语句中键入的位置有关。例如，假定有以下 SQL 语句：

```
SELECT EmployeeID FROM Employees as e WHERE e.EmployeeID>=20;
```

如果在 SELECT 后面打开文本完成窗口，则该列表将包含 Employees 表中的列名以及存储过程和 SQL 函数。

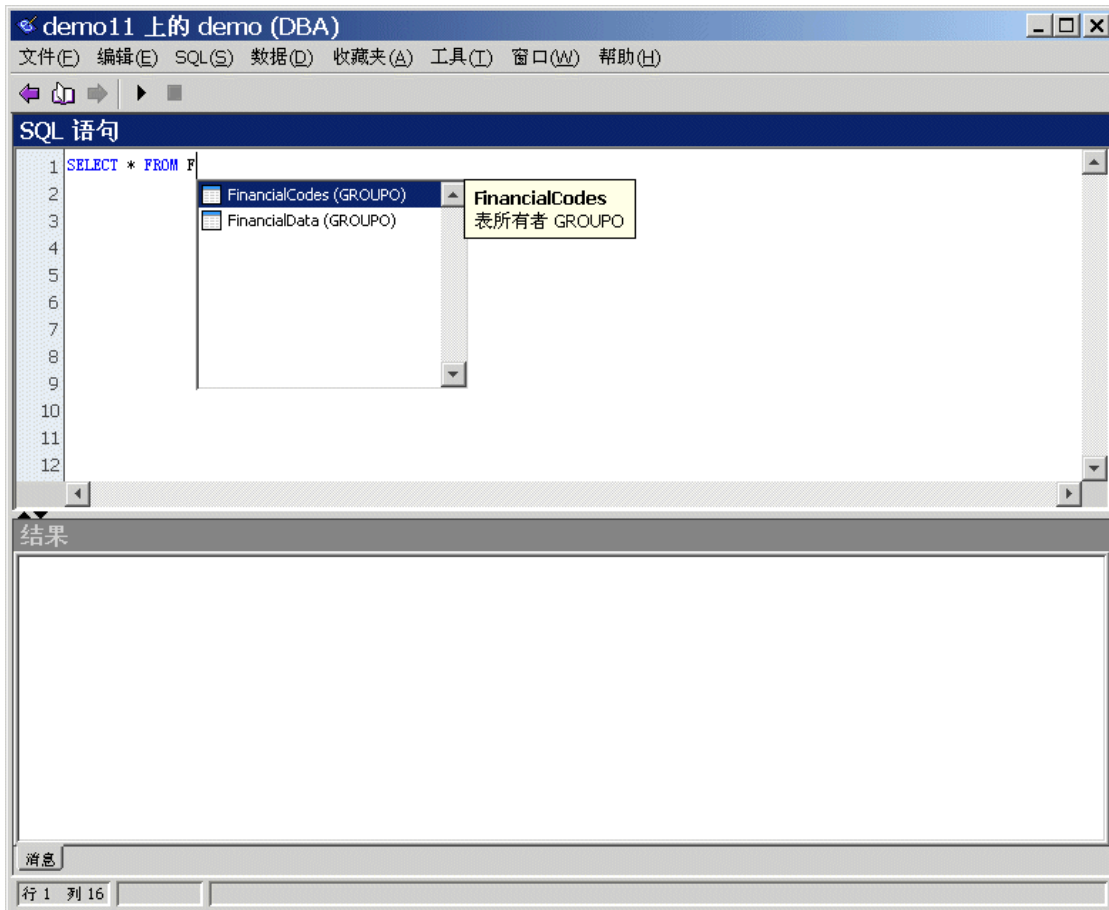
如果在 FROM 后面打开文本完成窗口，则该列表只包含表和存储过程。

如果在 WHERE 子句中的 e 后面打开文本完成窗口，则该列表只包含别名为 e 的表中的列。

◆ 使用文本完成

1. 在 Interactive SQL 中，在 [SQL 语句] 窗格中键入数据库对象名的第一个字母。
2. 按 Ctrl+Space 或 Ctrl+Shift+Space。

即会出现一个窗口，其中列出以所键入的字母开头的数据库对象名。在以下示例中，它会显示所有以字母 F 开头的数据库对象。



如果未看到想要的对象名，请按 Tab 键查看完整的数据库对象列表（取决于所设置的过滤选项—缺省情况下，所有数据库对象都会出现在列表中）。

3. 从列表中选择对象名，然后按 Enter 键。
对象名即会出现在 [SQL 语句] 窗格中。

可以从 Interactive SQL 中的 [选项] 窗口配置文本完成设置，也可在 Sybase Central 中的文本编辑器窗口上进行配置。

文本完成键盘快捷方式

在文本完成列表打开时可使用以下键盘快捷方式：

按键	说明
Ctrl+C	只在文本完成列表中显示列。

按键	说明
Ctrl+F	只在文本完成列表中显示 SQL 函数。
Ctrl+P	只在文本完成列表中显示存储过程和函数。
Ctrl+S	更改列表内容以显示或隐藏系统对象。
Ctrl+Shift+Space	打开文本完成窗口。也可使用 Ctrl+Space 打开文本完成窗口。
Ctrl+T	只在文本完成列表中显示表。
Ctrl+V	只在文本完成列表中显示视图。
Escape	在不添加任何文本的情况下关闭文本完成窗口。
Tab	在所有数据库对象列表与匹配目前为止已键入字母的名称列表之间切换。
*	对于表，插入以逗号分隔的列（包括数据类型）的列表。 对于存储过程，插入过程名，其后跟有以逗号分隔的参数名及其数据类型的列表。
+	对于表，插入以逗号分隔的列的列表。 对于存储过程，插入过程名，其后跟有以逗号分隔的参数名列表。
"	无论 <code>quoted_identifier</code> 选项的设置为何，都会完成名称，同时将其括在引号中。请参见“ quoted_identifier 选项 [兼容性] ”一节第 537 页。

使用快速启动程序选项

使用快速启动程序选项可缩短 Sybase Central 和 Interactive SQL 的启动时间。启用快速启动时，程序在关闭后会在内存中停留一段可配置的时间。如果您在这段时间内重新启动程序，程序会快速启动。如果在这段时间内未重新启动程序，则该进程会终止，并将其资源释放给操作系统。快速启动只在 Windows 上可用。

配置快速启动程序选项

快速启动程序选项使用计算机上的一个 TCP/IP 端口。如果另一程序已在使用此端口，可更改快速启动程序所使用的端口号。

如果在非活动状态计时器中指定的时间内未使用快速启动程序选项，它将结束并释放内存以供其它应用程序使用。缺省情况下，非活动状态计时器设置为 30 分钟。

◆ 配置 Interactive SQL 快速启动程序选项

1. 打开 Interactive SQL。
2. 选择 [工具] » [选项]。
3. 在左窗格中，单击 [常规]。
4. 单击 [配置]。
5. 完成 [端口号] 和 [关闭快速启动程序] 字段。
6. 单击 [确定]。
7. 单击 [确定]。

◆ 配置 Sybase Central 快速启动程序

1. 打开 Sybase Central。
2. 选择 [工具] » [选项]。
3. 在左窗格中，单击 [常规]。
4. 单击 [配置]。
5. 完成 [端口号] 和 [关闭快速启动程序] 字段。
6. 单击 [确定]。
7. 单击 [确定]。

使用 SQL Anywhere 控制台实用程序

SQL Anywhere 控制台实用程序为数据库服务器连接提供管理和监控工具。

有若干平台支持 SQL Anywhere 控制台实用程序。有关平台可用性，请参见 <http://www.sybase.com/detail?id=1062623>。

在不支持 SQL Anywhere 控制台实用程序的平台上，可以使用连接、数据库和数据库服务器属性来获取信息，也可以从一台运行支持 SQL Anywhere 控制台实用程序的操作系统（如 Windows、Mac OS X 或 Linux）的计算机上监控数据库服务器。

如果不具有 DBA 权限的用户连接到 SQL Anywhere 控制台实用程序，则所有需要 DBA 权限的功能都会被禁用。

有关 SQL Anywhere 控制台实用程序所支持的选项的信息，请参见“[SQL Anywhere 控制台实用程序 \(dbconsole\)](#)”一节第 780 页。

启动 SQL Anywhere 控制台实用程序

◆ 启动 SQL Anywhere 控制台实用程序（命令提示符）

- 运行以下命令：

```
dbconsole
```

如果未包括 `-c` 选项（该选项指定了数据库的连接参数）或所提供的连接参数不足，则会出现 **[连接]** 窗口，可在其中输入数据库的连接信息。

有关所支持选项的信息，请参见“[SQL Anywhere 控制台实用程序 \(dbconsole\)](#)”一节第 780 页。

以下命令会启动 SQL Anywhere 控制台实用程序并连接到示例数据库：

```
dbconsole -c "UID=DBA;PWD=sql;DSN=SQL Anywhere 11 Demo"
```

如果使用的是支持 Linux 桌面图标的 Linux 版本并且选择了在安装 SQL Anywhere 11 时安装它们，则可使用以下步骤。

◆ 启动 SQL Anywhere 控制台实用程序（Linux 桌面图标）

1. 从 **[应用程序]** 菜单中选择 **[SQL Anywhere 11]** » **[DBConsole]**
2. 在 **[连接]** 窗口中输入数据库的连接信息。
3. 单击 **[确定]**。

注意

以下步骤假定您已发起了运行 SQL Anywhere 实用程序的所需的脚本。请参见“[在 Unix 和 Mac OS X 上设置环境变量](#)”一节第 344 页。

◆ 启动 SQL Anywhere 控制台实用程序 (Unix 命令行)

1. 在终端会话中运行以下命令：

```
dbconsole
```

2. 在 [连接] 窗口中输入数据库的连接信息。
3. 单击 [确定]。

Mac OS X 注释

管理工具只能在使用 Apple JDK 1.6 所支持的 64 位处理器的 Intel Macintosh 机 (Mac OS X 10.5.2 或更高) 上运行。请参见 <http://www.sybase.com/detail?id=1062623>。

◆ 启动 SQL Anywhere 控制台实用程序 (Mac OS X)

1. 在 Finder 中，双击 `/Applications/SQLAnywhere11` 中的 [DBConsole]。
2. 在 [连接] 窗口中输入数据库的连接信息。

浏览 SQL Anywhere 控制台实用程序主窗口

SQL Anywhere 控制台实用程序由以下三个窗格组成：

- **连接** 显示有关当前数据库连接的信息。
- **属性** 显示有关当前运行的数据库和数据库服务器的信息。
- **消息** 显示数据库服务器消息。

可以使用 [选项] 窗口配置每个窗格中出现的信息。

◆ 自定义 [连接] 窗格的内容

1. 在 SQL Anywhere 控制台实用程序中，选择 [文件] » [选项]。
2. 在左窗格中，单击 [连接查看器]。
3. 选择想要显示在 [连接] 窗格内的属性。
4. 单击 [确定]。

◆ 自定义 [属性] 窗格的内容

1. 在 SQL Anywhere 控制台实用程序中，选择 [文件] » [选项]。
2. 在左窗格中，单击 [属性查看器]。
3. 选择想要在 [属性] 窗格中显示的数据库和数据库服务器属性。

4. 单击 [确定]。

◆ **自定义 [消息] 窗格的内容**

1. 在 SQL Anywhere 控制台实用程序中，选择 [文件] » [选项]。
2. 在左窗格中，单击 [消息查看器]。
3. 为显示在 [消息] 窗格中的消息选择消息选项。
4. 单击 [确定]。

检查软件更新

可对 SQL Anywhere 进行配置，使其在检测到更新（如 EBF 和维护版本）时通知您。缺省情况下，SQL Anywhere 不会检查软件更新。

自动检查更新

Sybase Central、Interactive SQL 和 SQL Anywhere 控制台实用程序 (dbconsole) 都提供了配置更新检查程序的方法，该程序控制 SQL Anywhere 是否应检查软件更新以及每隔多久就应检查一次。

◆ 配置更新检查程序 (Sybase Central)

1. 选择 [帮助] » [SQL Anywhere 11] » [配置更新检查程序]。
2. 编辑更新检查程序设置。
3. 单击 [确定]。

◆ 配置更新检查程序 (Interactive SQL)

1. 选择 [工具] » [选项]。
2. 在左窗格中，单击 [SQL Anywhere]。
3. 单击 [检查更新] 选项卡。
4. 编辑更新检查程序设置。
5. 单击 [确定]。

◆ 配置更新检查程序 (SQL Anywhere 控制台实用程序)

1. 选择 [文件] » [选项]。
2. 在左窗格中，单击 [检查更新]。
3. 编辑更新检查程序设置。
4. 单击 [确定]。

手工检查更新

随时都可以通过执行以下操作之一来检查 SQL Anywhere 软件更新：

- **[开始] 菜单** 选择 [开始] » [程序] » [SQL Anywhere 11] » [检查更新]。
- **Sybase Central** 选择 [帮助] » [SQL Anywhere 11] » [检查更新]。
- **Interactive SQL** 选择 [帮助] » [检查更新]。
- **SQL Anywhere 控制台实用程序 (dbconsole)** 选择 [帮助] » [检查更新]。
- **SQL Anywhere 支持实用程序 (dbsupport)** 发出以下命令：

```
dbsupport -iu
```

- **Sybase Web 站点** 转到 <http://downloads.sybase.com>。

另请参见

- “SQL Anywhere 中的错误报告” 一节第 76 页
- “支持实用程序 (dbsupport)” 一节第 785 页

数据库管理实用程序

目录

管理实用程序概述	695
备份实用程序 (dbbackup)	698
广播转发器实用程序 (dbns11)	703
证书创建实用程序 (createcert)	705
证书查看器实用程序 (viewcert)	708
数据源实用程序 (dbdsn)	710
dbisqlc 实用程序 (不建议使用)	720
消除实用程序 (dberase)	722
文件隐藏实用程序 (dbfhide)	724
直方图实用程序 (dbhist)	726
信息实用程序 (dbinfo)	728
初始化实用程序 (dbinit)	729
Interactive SQL 实用程序 (dbisql)	740
密钥对生成器实用程序 (createkey)	744
语言选择实用程序 (dblang)	745
日志传送管理器实用程序 (dbltm)	748
翻译日志文件实用程序 (dbtran)	753
Ping 实用程序 (dbping)	757
重建实用程序 (rebuild)	760
脚本执行实用程序 (dbrunsql)	761
服务器枚举实用程序 (dblocate)	763
服务器授权实用程序 (dblic)	766
用于 Linux 的服务实用程序 (dbsvc)	769
用于 Windows 的服务实用程序 (dbsvc)	773
SQL Anywhere 控制台实用程序 (dbconsole)	780
后台启动服务器实用程序 (dbspawn)	781
停止服务器实用程序 (dbstop)	783
支持实用程序 (dbsupport)	785
事务日志实用程序 (dblog)	792
卸载实用程序 (dbunload)	795
升级实用程序 (dbupgrad)	808

校验实用程序 (dbvalid)	810
版本诊断实用程序 (dbversion)	813

管理实用程序概述

SQL Anywhere 包括一组用于执行数据库管理任务的实用程序。可以从一个或多个 Sybase Central、Interactive SQL 或在命令提示符处访问每个实用程序。

有关平台可用性，请参见 <http://www.sybase.com/detail?id=1062617>。

管理实用程序会使用一组注册表条目或 .ini 文件。请参见“注册表和 INI 文件”一节第 373 页。

数据库文件管理语句

可以使用一组 SQL 语句来执行由管理实用程序执行的某些任务。请参见“SQL 语句”《SQL Anywhere 服务器 - SQL 参考》。

另请参见

- “使用 Sybase Central”一节第 620 页
- “使用 Interactive SQL”一节第 636 页

使用配置文件

随 SQL Anywhere 提供的许多实用程序都允许将命令行选项存储在一个配置文件中。如果使用的选项较多，您会发现将它们存储在一个配置文件中是非常有用的。

@data 选项用于在命令行上指定环境变量和配置文件。要指定配置文件，请用配置文件的路径和名称替换 data。如果存在和配置文件同名的环境变量，则使用环境变量。

配置文件可以包含换行符，也可以包含任何一组选项（包括 @data 选项）。可以使用井号 (#) 指定注释行。在行尾单独出现的和号 (&) 字符表示在下一行继续前一个标识。例如，下面的配置文件可用于启动镜像服务器：

```
-n server1
-o server1.conslog
-gd all
-su sql
-hs
-x tcpip(port=2638;dobroadcast=no)
-xf server1.state
asatest.db
-sn asatest
-xp partner=(eng=server2;links=tcpip(port=2637;timeout=1)); &
  arbiter=(eng=arbiter;links=tcpip(port=2639;timeout=1)); &
  mode=sync; &
  auth=abc
```

@data 参数可以出现在命令行中的任何位置，文件中所含的参数会插入到该位置。在一个命令行中可以多次使用 @data 选项来指定多个配置文件。

实用程序通过展开指定的配置文件并从左到右读取整个命令行的方式来读取命令行。如果您指定的选项能被命令行中的其它选项替换，则更靠近命令行末尾的选项会替换前面的选项。在某些情况下，冲突的选项会引发错误。

注意

后台启动服务器实用程序 (dbspawn) 不会展开由 @data 选项指定的配置文件。

如果要保护口令或配置文件中的其它信息，您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。

有关配置文件内容模糊处理的详细信息，请参见“文件隐藏实用程序 (dbfhide)”一节第 724 页。

示例

下面的配置文件为校验实用程序 (dbvalid) 保存一组选项：

```
#Connect to the sample database as the user DBA with password sql
-c "UID=DBA;PWD=sql;DBF=samples-dir\demo.db"
#Perform an express check on each table
-fx
#Log output messages to the specified file
-o "c:\validationlog.txt"
```

有关 *samples-dir* 的信息，请参见“示例目录”一节第 368 页。

如果此配置文件保存为 *c:\config.txt*，则在命令中可按如下方式使用：

```
dbvalid @c:\config.txt
```

在配置文件中使用条件分析

在配置文件中可以使用条件分析来指定哪些实用程序可以使用该文件。通过条件指令可以根据使用该文件的实用程序来包括或排除命令参数。在配置文件中使用时，仍可使用文件隐藏实用程序 (dbfhide) 来隐藏该文件的内容。

语法

configuration-file = *text*...

text : *comment* | *conditional* | *command-line-option*

comment : *line starting with # that is not a conditional*

conditional :

```
#if condition
text
[ #elif condition
text
] ...
[ #else
text
] ...
#endif
```

condition : { *tool=utility-name*[,*utility-name*]... | *utility-name* }

utility-name 可以是以下值：

dbbackup	dbinfo	dbltn	dbstop	dbxtract
dbdsn	dbinit	dbmlsync	dbsupport	mlsrv
dbeng	dblic	dbping	dbsvc	mluser
dberase	dblocate	dbremote	dbunload	qaagent
dbfhide	dblog	dbspawn	dbupgrad	rteng
dbhist	dblsn	dbsrv	dbvalid	

用法

要成为条件指令，命令行中的第一个非空白字符必须是#。如果在#if或#elif指令中指定了某个实用程序，该指令之后、下一条条件指令之前的所有行都会包含在此实用程序的命令行中。#else指令用于处理之前条件块中未处理的实用程序的情况。#endif指令是条件指令结构的结束。

由**tool=**指定的工具名列表中不允许出现空格。条件指令可以嵌套。如果在分析配置文件的过程中出现错误，实用程序会报告无法打开配置文件。

示例

以下配置文件可由dbping、dbstop和dbvalid使用。

```
#if tool=dbping,dbstop,dbvalid
#always make tools quiet
-q
-c "UID=DBA;PWD=sql;ENG=myserver;DBN=mydb"
#if dbping
#make a database connection
-d
#elif tool=dbstop
#don't ask
-y
#else
#must be dbvalid
#use WITH EXPRESS CHECK
-fx
#endif
#endif
```

备份实用程序 (dbbackup)

为运行中的数据库创建数据库文件和事务日志的客户端或服务器端备份。

语法

dbbackup [*options*] *target-directory*

选项	说明
@ <i>data</i>	<p>从指定的环境变量或配置文件读入选项。请参见“使用配置文件”一节第 695 页。</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“文件隐藏实用程序 (dbfhide)”一节第 724 页。</p>
-b <i>block-size</i>	<p>指定在将页面从数据库服务器转移到 dbbackup 时所用的最大块大小（以页数计）。dbbackup 实用程序会尝试分配此页数；如果分配失败，它会将该值减半，然后再次尝试，直到分配成功。缺省大小为 128 页。</p>
-c " <i>keyword=value; ...</i> "	<p>指定连接参数。用户 ID 必须有 DBA 或 REMOTE DBA 权限才能连接到该数据库。请参见“连接参数”一节第 248 页。</p> <p>例如，以下命令会将运行于 <code>sample_server</code> 数据库服务器（作为 DBA 用户连接）上的示例数据库备份到 <code>SQLAnybackup</code> 目录中：</p> <pre>dbbackup -c "ENG=sample_server;DBN=demo;UID=DBA;PWD=sql" SQLAnybackup</pre>
-d	<p>只备份主数据库文件，不备份事务日志文件（即使存在）。</p>

选项	说明
<p>-k <i>checkpoint-log-copy-option</i></p>	<p>指定 dbbackup 在将数据库文件写入目标目录之前对数据库文件的处理方式。选择在备份期间是应用前映像还是复制检查点日志会对性能产生影响。如果通过指定 -s 选项在服务器上执行备份，则 -k 的缺省设置为 auto；否则，缺省设置为 copy。</p> <p>auto 数据库服务器会检查备份目录所在卷的可用磁盘空间量。备份开始时，如果可用磁盘空间至少是数据库大小的两倍，备份工作会像指定了 copy 那样继续进行。否则，会像指定了 nocopy 那样继续进行。只有指定了 -s 才可以使用此设置。</p> <p>copy 备份期间会读取数据库文件，但不对任何修改过的页面应用前映像。整个检查点日志以及系统 dbspace 会复制到备份目录中。下次数据库启动时，数据库服务器会自动将其恢复到备份开始时在检查点所处的状态。</p> <p>因为页面前映像无需写入临时文件，所以使用此选项可以实现更好的备份性能，并能减少服务器内部对备份期间正在运行的其它连接的争用。但由于数据库文件的备份副本包含检查点日志，而后者又包含备份开始之后所修改的任何页面的前映像，所以数据库文件的备份副本可能要比备份开始时的数据库文件大一些。copy 选项应在目标目录不存在磁盘空间问题的情况下使用。</p> <p>nocopy 备份期间则不会复制检查点日志。如果选择此选项，曾修改过的页面的前映像会被保存在临时文件中，以便在备份过程中加以应用。数据库文件的备份副本与备份操作开始时的数据库大小相同。由于备份副本中不含检查点日志，因此实际可能会略小一些。使用此选项可以减小备份的数据库文件，但会减缓备份过程，并有可能降低数据库服务器中其它操作的性能。它在目标驱动器空间有限的情况下非常有用。</p> <p>recover 数据库服务器会复制检查点日志（与使用 copy 选项时一样），但会在备份结束时将检查点日志应用于数据库。这样会将备份的数据库文件恢复到备份操作开始时的状态（和大小）。此选项在备份驱动器空间有限的情况下非常有用（它用来备份检查点日志的空间量与 copy 选项相同，但生成的文件会小一些）。只有指定了 -s 才可以使用此设置。</p>
<p>-l <i>filename</i></p>	<p>可实现在服务器崩溃时使辅助系统快速启动。活动备份不会停止。它会在服务器运行期间持续运行。它会一直运行，直到主服务器崩溃。主服务器崩溃时，活动备份会关闭，但备份的日志文件保持原状，并可以用于快速启动辅助系统。请参见“活动备份和事务日志镜像之间的区别”一节第 822 页和“创建活动备份”一节第 830 页。</p> <p>如果指定 -l，则无法使用 -s 在服务器上创建映像备份。</p>

选项	说明
-n	<p>此选项将备份事务日志文件的命名约定更改为 <i>yymmddxx.log</i>，其中 <i>xx</i> 是从 AA 到 ZZ 的连续字母，<i>yymmdd</i> 代表当前的年月日。此选项与 -r 一起使用。</p> <p>事务日志文件的备份副本会存储在命令所指定的目录中，并采用 <i>yymmddxx.log</i> 命名约定。这样，多个版本的事务日志文件备份副本就可以保存在同一个备份目录中。</p> <p>也可以使用 -x 和 -n 选项来重命名日志副本。例如</p> <pre>dbbackup -c "UID=DBA;PWD=sql" -x -n mybackupdir</pre>
-o filename	将输出消息写入指定的文件中。
-q	不显示输出消息。只有在命令提示符下运行此实用程序时才可以使用此选项。
-r	<p>重命名事务日志，并启动新的事务日志。它会强制执行检查点操作并导致执行以下三个步骤：</p> <ol style="list-style-type: none"> 1. 复制当前的工作事务日志文件并将其保存到在命令中指定的目录。 2. 当前的事务日志仍然在其当前的目录中，但其名称已改为 <i>yymmddxx.log</i> 形式，其中 <i>xx</i> 是从 AA 到 ZZ 的连续字符，<i>yymmdd</i> 代表当前的年月日。于是，此文件就不再是当前事务日志了。 3. 生成一个不包含事务的新事务日志文件。其名称与先前被视为当前事务日志的文件的名称相同，数据库服务器用它来作为当前事务日志。 <p>使用数据库镜像时，请不要使用此选项。请参见“数据库镜像和事务日志文件”一节第 901 页。</p>
-s	使用 BACKUP DATABASE 语句在服务器上创建映像备份。如果指定 -s 选项，就无法使用 -l 选项（创建事务日志的活动备份）。指定的目录相对于服务器的当前目录，因此建议指定完整的路径名。另外，服务器在指定的目录上必须具有写权限。指定 -s 后，备份实用程序不显示进度消息，且在其覆盖现有文件时不进行提示。如果希望在尝试覆盖现有文件时获得提示，请不要指定 -s 或 -y 。如果指定 -k 恢复选项，就必须指定 -s 。
-t	创建用作增量备份的备份，因为事务日志可应用于数据库文件的最新备份副本。

选项	说明
-x	<p>备份现有事务日志，删除原始日志，然后启动新事务日志。使用数据库镜像时，请不要使用此选项。请参见“数据库镜像和事务日志文件”一节第 901 页。</p> <p>小心 使用此选项会导致数据库无法从介质故障中恢复。仅在允许数据丢失的情况下，才可使用此选项。</p>
-xo	删除当前事务日志并启动新的事务日志。此操作不执行备份；其目的是在非复制环境中释放磁盘空间。使用数据库镜像时，请不要使用此选项。请参见“ 数据库镜像和事务日志文件 ”一节第 901 页。
-y	创建备份目录或替换目录中以前的备份文件，而无需确认。如果希望在尝试覆盖现有文件时得到提示，请不要指定 -s 或 -y。
<i>target-directory</i>	指定将备份文件复制到其中的目录。如果该目录不存在，则会进行创建。但是，父目录必须存在。缺省情况下，备份实用程序会创建一个数据库文件的客户端备份。您可以使用 BACKUP DATABASE 语句，通过指定 -s 在服务器上创建备份。

注释

备份实用程序会为单个数据库的所有文件创建一个备份副本。一个简单的数据库由以下两个文件组成：主数据库文件和事务日志。更复杂的数据库可以将表存储在多个文件中，每个文件都是一个单独的 dbspace。所有备份文件都与数据库文件同名。备份实用程序创建的映像备份由每个所备份文件对应的单独文件组成。

有关创建档案备份（包含数据库文件和事务日志的单个文件）的详细信息，请参见“[档案备份](#)”一节第 823 页。

对运行中的数据库使用备份实用程序等同于在数据库未运行时复制数据库文件。您可以在其它应用程序或用户使用数据库时，使用备份实用程序来备份它。

如果 -d 或 -t 选项均未使用，则备份所有的数据库文件。

缺省情况下，备份实用程序会创建一个数据库文件的客户端备份。您可以使用 BACKUP DATABASE 语句，通过指定 -s 在服务器上创建备份。

有关执行服务器端备份的信息，请参见“[BACKUP 语句](#)”一节《[SQL Anywhere 服务器 - SQL 参考](#)》。

小心

不得以任何方式更改数据库和事务日志的备份副本。如果在备份过程中未执行任何事务，或者指定了 `BACKUP DATABASE WITH CHECKPOINT LOG RECOVER` 或 `WITH CHECKPOINT LOG NO COPY`，则可以使用只读模式或通过验证备份数据库的副本来验证备份数据库的有效性。

但是，如果有正在执行的事务，或者指定了 `BACKUP DATABASE WITH CHECKPOINT LOG COPY`，则当您启动数据库服务器时，数据库服务器必须执行数据库恢复。恢复功能会修改备份副本，这是不希望出现的情况。

除了 `dbbackup` 外，还可以采用下列方式访问备份实用程序：

- 从 Sybase Central 使用 [\[创建备份映像向导\]](#)。请参见“映像备份”一节第 823 页。
- 从 Interactive SQL 使用 `BACKUP DATABASE` 语句。请参见“[BACKUP 语句](#)”一节《[SQL Anywhere 服务器 - SQL 参考](#)》。

有关所推荐的备份过程的详细信息，请参见“[备份和数据恢复](#)”第 817 页。

退出代码是零（成功）或非零（失败）。

有关退出代码的详细信息，请参见“[软件组件的退出代码](#)”一节《[SQL Anywhere 服务器 - 编程](#)》。

广播转发器实用程序 (dbns11)

允许 SQL Anywhere 客户端穿过防火墙查找运行于其它子网中、UDP 广播通常无法到达的 SQL Anywhere 数据库服务器。

语法

dbns11 [*options*] [*address ...*]

选项	说明
@data	读取来自指定的环境变量或配置文件的选项。请参见“使用配置文件”一节第 695 页。 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“文件隐藏实用程序 (dbfhide)”一节第 724 页。
-ap port	指定数据库服务器所用的端口号。缺省端口号为 2638。
-m ip	指定正在运行该 DBNS 进程的计算机的 IP 地址。对于使用多个 IP 地址的计算机，此参数是必须要指定的。该地址必须是 IPv4 地址。
-o filename	将广播转发器消息窗口中出现的输出内容写入指定的文件中。
-p port	指定 DBNS 广播转发器所用的端口号。缺省值为 3968。除需要打开端口 2638 以实现标准的客户端-服务器通信外，如果子网之间有防火墙，还必须打开广播转发器实用程序所用的端口号，以便 DBNS 进程之间进行 TCP 连接。
-q	以安静模式运行—不显示消息。
-s	让新 DBNS 进程检查是否有另一个 DBNS 进程已在该子网中运行，如果发现另一个 DBNS 进程，则先返回一个错误，然后关闭。
-x host	关闭在指定主机上运行的 DBNS 进程。可以指定 IP 地址或主机名。
-z	在调试模式下启动 DBNS 广播转发器。以调试模式运行时，对于每个接收或转发的 SQL Anywhere 广播包，广播转发器消息窗口中都会出现一行消息。由于调试过程中会输出大量消息，因此只有在出现连接问题时才应使用调试模式。
<i>address</i>	指定正在或即将运行 DBNS 进程的其它计算机的 IP 地址或主机名。这样，DBNS 进程就可以检测到彼此，并交换有关已知数据库服务器和其它 DBNS 进程的信息。

注释

允许 SQL Anywhere 客户端穿过防火墙查找运行于其它子网中、UDP 广播通常无法到达的 SQL Anywhere 数据库服务器，而无需使用 HOST 连接参数或 LDAP。

address 可以是 IP 地址，也可以是计算机名。多个地址用空格分隔。

此实用程序可在受支持的 Unix 平台以及所有 32 位和 64 位 Windows 平台上使用。

客户端和数据库服务器必须运行 SQL Anywhere 9.0.2 或更新版本才能使用广播转发器。

小心

建议不要在 SQL Anywhere 数据库服务器所在的计算机上运行 dbns11 实用程序，因为这样 dbns11 或数据库服务器可能无法接收到 UDP 广播。

另请参见

- [“使用广播转发器实用程序查找数据库服务器”一节第 128 页](#)

示例

假设您希望允许子网 10.50.83.255 和 10.50.125.255 中的计算机使用广播进行连接。此时需要 10.50.83.255 和 10.50.125.255 子网中各有一台计算机，分别为计算机 A（地址为 10.50.83.114）和计算机 B（地址为 10.50.125.103）。

在这两台计算机上运行 dbns11，以另一计算机的 IP 地址为参数。在计算机 A 上执行以下命令：

```
dbns11 10.50.125.103
```

在计算机 B 上执行以下命令：

```
dbns11 10.50.83.114
```

如果其中一台计算机有多个 IP 地址，则还必须使用 -m 选项指定本地 IP 地址。例如，在计算机 A 上应使用以下命令：

```
dbns11 -m 10.50.83.114 10.50.125.103
```

证书创建实用程序 (createcert)

创建 X.509 证书。

语法

`createcert [-r | -s]`

选项	说明
<code>-r</code>	创建 PKCS10 证书请求。如果指定此选项， <code>createcert</code> 不会提示输入签署者或用于签署证书的任何其它信息。
<code>-s filename</code>	签署指定文件中的 PKCS10 证书请求。该请求可以是采用 DER 或 PEM 编码的。如果指定此选项， <code>createcert</code> 不会提示输入密钥生成或主题信息。

注释

用户通常会向第三方购买证书。这些证书颁发机构提供其自己用于创建证书的工具。以下工具会对创建用于开发和测试目的的证书十分有用，但也可以用于创建生产证书。

要创建签名证书，应使用不带选项的 `createcert`。如果要将这一过程分解为两个步骤，例如由一个人创建请求，而由另一个人签署请求，则第一个人可以运行带 `-r` 的 `createcert` 来创建请求，而第二个人可以通过运行带 `-s` 的 `createcert` 来签署该请求。

运行 `createcert` 时，系统会提示您输入以下信息。如果指定 `-r` 或 `-s` 选项，其中某些提示则不会显示。

- **选择加密类型** 只有在已购买了用于 ECC 加密的许可时，此提示才会显示。选择 RSA 或 ECC。
- **输入 RSA 密钥长度 (512-16384)** 只有选择了 RSA 加密，此提示才会显示。可以选择一个介于 512 位到 16384 位之间的长度。
- **输入 ECC 曲线** 只有在已购买了用于 ECC 加密的许可而且选择了上面的 ECC 加密类型时，此提示才会显示。系统会提示您从 ECC 曲线列表中进行选择。缺省值为 `sect163k1`。
- **主题信息** 您必须输入以下标识实体的信息：
 - 国家/地区代码
 - 省/直辖市/自治区
 - 地点
 - 组织
 - 组织单位
 - 公用名
- **输入签署者的证书的文件路径** 或者提供签署者的证书的位置和文件名。如果提供此信息，所生成的证书就是一个签名证书。如果未提供此信息，所生成的证书就是一个自签名根证书。
- **输入签署者的专用密钥的文件路径** 提供用于保存与证书请求关联的专用密钥的位置和文件名。只有在前一个提示中提供一个文件时，才会显示此提示。

- **输入签署者的专用密钥的口令** 提供加密签署者的专用密钥时所使用的口令。只有在加密专用密钥时才提供此口令。
- **序列号** 或者提供序列号。序列号必须为不长于 40 位的十六进制字符串。此数字在当前签署者签署的所有证书中必须是唯一的。如果未提供序列号，`createcert` 将生成一个 GUID 作为序列号。
- **证书有效年数 (1-100)** 指定证书的有效年数（介于 1 和 100 之间）。在此期限后，该证书及其签署的所有证书都将到期。
- **是 (y) 否 (n) 是证书颁发机构** 指示此证书是否可用于签署其它证书。缺省情况下，证书不是证书颁发机构 (n)。
- **密钥用法** 提供一个以逗号分隔的数字列表，指示如何使用证书的专用密钥。这是一个高级选项；大多数情况下应可接受缺省值。该缺省值取决于证书是否是证书颁发机构。
- **保存请求的文件路径** 只有指定 `-r` 选项，此提示才会显示。提供 PKCS10 证书请求的位置和文件名。
- **输入文件路径以保存证书** 提供保存证书的位置和文件名。除非指定位置和文件名，否则证书不会被保存。
- **输入文件路径以保存专用密钥** 提供保存专用密钥的位置和文件名。
- **输入口令以保护专用密钥** 或者提供用来加密专用密钥的口令。如果不提供口令，则专用密钥不会被加密。只有在前一个提示中提供一个文件时，才会显示此提示。
- **输入文件路径以保存标识** 提供保存标识的位置和文件名。标识文件是证书、签署者和专用密钥的连接。这是您在启动时向服务器提供的文件。如果未保存专用密钥，`createcert` 将提示输入口令来保存专用密钥。否则将使用以前提供的口令。除非提供文件名，否则标识不会被保存。如果不保存标识文件，可将证书、签署者和专用密钥文件手工连接到一个标识文件中。

另请参见

- [“证书”一节第 1032 页](#)
- [“证书查看器实用程序 \(viewcert\)”一节第 708 页](#)
- [“-ec 服务器选项”一节第 171 页](#)
- [“Encryption 连接参数 \[ENC\]”一节第 265 页](#)
- [“FIPS 认可的加密技术”一节第 1033 页](#)

示例

以下示例将创建一个签名证书。在该示例中，没有为签署者的证书提供文件名，这使其成为一个自签名根证书。

```
>createcert
SQL Anywhere X.509 证书生成器版本 11.0.1.3330
选择加密类型 ((R)SA 或 (E)CC): r
输入 RSA 密钥长度 (512-16384): 1024
正在生成密钥对...
国家/地区代码: CA
省/直辖市/自治区: Ontario
地点: Waterloo
组织: Sybase iAnywhere
组织单位: Engineering
公用名: Test Certificate
```

```
输入签署者的证书的文件路径:
证书将是自签名根
序列号 [生成 GUID]:
生成的序列号: bfb89a26fb854955954cab4d056e177
证书有效年数 (1-100): 10
证书颁发机构 (Y/N) [N]: n
1. 数字签名
2. 不可否认性
3. 密钥加密
4. 数据加密
5. 密钥协议
6. 证书签名
7. CRL 签名
8. 仅加密
9. 仅解密
密钥用法 [3,4,5]: 3,4,5
输入文件路径以保存证书: cert.pem
输入文件路径以保存专用密钥: key.pem
输入口令以保护专用密钥: pwd
输入文件路径以保存标识: id.pem
```

要生成企业根证书（用于签署其它证书的证书），必须使用证书颁发机构创建自签名根证书。其过程类似于上述过程。但是，对于证书颁发机构提示的答复应为 **[yes]**，而且对于角色的选择应为选项 **6,7**（缺省值）。

```
证书颁发机构 (Y/N) [N]: y
1. 数字签名
2. 不可否认性
3. 密钥加密
4. 数据加密
5. 密钥协议
6. 证书签名
7. CRL 签名
8. 仅加密
9. 仅解密
密钥用法 [6,7]: 6,7
```

证书查看器实用程序 (viewcert)

显示公共密钥基础结构 (Public Key Infrastructure, 简称 PKI) 对象内的值、转换 PKI 对象的编码, 或者加密及解密专用密钥。

语法

viewcert [*options*] *input-file*

选项	说明
-d	对输出进行 DER 编码。此选项只能与 -o 选项一起使用。而不能与 -p 一起使用。缺省情况下, viewcert 以可读文本的格式输出 PKI 对象。
-ip <i>input-password</i>	如果 <i>input-file</i> 中包含加密的专用密钥, 使用此选项指定对其解密所需的口令。
-o <i>output-file</i>	指定 viewcert 用来写入输出内容的文件。缺省情况下, viewcert 会将输出写入到它运行时所在的命令提示窗口。
-op <i>output-password</i>	指定 viewcert 加密专用密钥所用的口令。此选项只能与 -d 或 -p 一起使用。缺省情况下, 专用密钥不加密。
-p	对输出进行 PEM 编码。此选项只能与 -o 选项一起使用。而不能与 -d 一起使用。缺省情况下, viewcert 以可读文本的格式输出 PKI 对象。
<i>input-file</i>	指定 必须是 DER 或 PEM 编码 PKI 对象的文件。

注释

viewcert 实用程序可用于查看以下类型的 PKI 对象:

- X.509 证书
- 证书请求
- 专用密钥
- 证书撤销列表 (CRL)

Viewcert 还可用于在 DER 和 PEM 编码类型之间进行转换, 以及加密或解密专用密钥。

viewcert 实用程序支持 RSA 和 ECC 对象。要查看 ECC 对象, 您必须订购一个单独的许可。请参见“[单独授权的组件](#)”一节 《[SQL Anywhere 11 - 简介](#)》。

另请参见

- “[证书创建实用程序 \(createcert\)](#)” 一节第 705 页

示例

以下示例将供您查看随 SQL Anywhere 提供的示例 RSA 证书:

```
viewcert rsaroot.crt
```

此示例将产生以下输出:

```
SQL Anywhere X.509 证书查看器版本 11.0.1.3330
```

```
X.509 证书
```

```
-----  
公用名: RSA Root  
组织单位: test  
组织: test  
地点: test  
省/直辖市/自治区: test  
国家/地区代码: test  
发行者: RSA Root  
序列号: 303031  
发行: Apr 15, 2002 12:53:51  
到期: Apr 16, 2022 12:53:51  
签名算法: RSA、MD5  
密钥类型: RSA  
密钥大小: 1024 位  
基本约束: 是证书颁发机构, 路径长度限制: 10  
密钥用法: 证书签名、CRL 签名
```

数据源实用程序 (dbdsn)

创建、删除、说明和列出 SQL Anywhere ODBC 数据源。

语法

```
dbdsn [ modifier-options ]
  { -l[ s | u ]
  | -d[ s | u ] dsn
  | -g[ s | u ] dsn
  | -w[ s | u ] dsn [ details-options;... ]
  | -cl }
```

主选项	说明
@data	读取来自指定的环境变量或配置文件的选项。请参见“ 使用配置文件 ”一节第 695 页。 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“ 文件隐藏实用程序 (dbfhide) ”一节第 724 页。
-l[s u]	列出可用的 SQL Anywhere ODBC 数据源。您可以使用 -b 或 -v 选项修改列表格式。在 Windows 中，可以使用 u (用户) 或 s (系统) 指定符来修改该选项。缺省指定符是 u 。
-d[s u] dsn	删除指定的 SQL Anywhere 数据源。如果提供 -y，则删除任何现有数据源而无需确认。在 Windows 中，可以使用 u (用户) 或 s (系统) 指定符来修改该选项。缺省指定符是 u 。
-g[s u] dsn	列出指定 SQL Anywhere 数据源的定义。您可以使用 -b 或 -v 选项修改输出的格式。在 Windows 中，可以使用 u (用户) 或 s (系统) 指定符来修改该选项。缺省指定符是 u 。
-w[s u] dsn [details-options]	创建新数据源；如果存在同名的数据源，则覆盖它。如果提供 -y，则覆盖任何现有数据源而无需确认。在 Windows 中，可以使用 u (用户) 或 s (系统) 指定符来修改该选项。缺省指定符是 u 。
-cl	列出 dbdsn 实用程序支持的连接参数。请参见“ 连接参数 ”一节第 248 页。 有关支持的 ODBC 连接参数的信息，请参见“ ODBC 连接参数 ”一节第 713 页。

修饰符选项	说明
-b	将列表的输出格式设置为单行连接字符串。
-cm	<p>显示用来创建数据源的命令。此选项可用于将创建命令输出到一个文件中。可使用该创建命令将数据源添加到其它计算机，或者如果数据源有更改，也可用该命令将数据源恢复到其原始状态。您必须使用 -cm 指定 -g 选项或 -l 选项，否则命令将失败。指定 -g 会显示指定数据源的创建命令，而指定 -l 则会显示所有数据源的创建命令。</p> <p>如果指定的数据源不存在，则会生成删除数据源的命令。例如，如果计算机上不存在 mydsn 数据源，则 <code>dbdsn -cm -g mydsn</code> 会返回下列命令来删除 mydsn 数据源：</p> <pre>dbdsn -y -du "mydsn"</pre>
-dr	<p>显示数据源时包括 Driver 参数。在使用 -cm 选项重新创建数据源时，它允许当前版本的 dbdsn 创建那些引用另一个版本的 ODBC 驱动程序的数据源。</p> <p>例如，假设您使用以下命令来创建版本 9 的数据源：</p> <pre>dbdsn -y -wu "9.0 Student Sample" -c "UID=DBA;PWD=sql;...;Driver=Adaptive Server Anywhere 9.0"</pre> <p>执行 <code>dbdsn -cm -l</code> 时，dbdsn 列出不包括 Driver= 参数的相同命令，随后可使用 SQL Anywhere 版本 10.0 ODBC 驱动程序来重新创建 DSN。</p> <p>但如果执行 <code>dbdsn -dr -cm -l</code>，Driver= 参数会包括在内，数据源会完全按照最初的创建方式重新创建：即使用版本 9 ODBC 驱动程序。</p>
-f	显示所使用的系统文件的名称。此选项只能在 Unix 中使用。

修饰符选项	说明
-ns	<p>指定用于确定系统信息文件（缺省情况下名为 <code>.odbc.ini</code>）位置的环境变量设置。如果在环境中存在多个候选系统信息文件，则此选项对于确定 <code>dbdsn</code> 正在使用哪个文件非常有用。此选项只能在 Unix 中使用。</p> <p>如果在创建数据源时不指定 <code>-ns</code>，<code>dbdsn</code> 也会在用户主目录和路径中检查系统信息文件。</p> <p>有关如何查找系统信息文件的详细信息，请参见“在 Unix 上使用 ODBC 数据源”一节第 95 页。</p>
-o filename	<p>将输出消息写入指定的文件中。</p>
-or	<p>用 <code>-c</code> 选项指定时，可创建 iAnywhere Solutions Oracle 驱动程序的数据源。例如：</p> <pre data-bbox="785 801 1324 879">dbdsn -w MyOracleDSN -or -c Userid=DBA;Password=sql; SID=abcd;ArraySize=500;ProcResults=y</pre> <p>您可以指定 <code>-cl</code> 选项和 <code>-or</code> 选项，以获得 iAnywhere Solutions Oracle 驱动程序的连接参数列表。</p> <p>有关详细信息，请参见“iAnywhere Solutions Oracle 驱动程序”一节《MobiLink - 服务器管理》。</p>
-pe	<p>指定此选项且 DSN 中包括 PWD 条目时，将 PWD 条目中的密码加密，并将 PWD 条目替换为包含已加密密码的 ENP 条目。</p>
-q	<p>取消向数据库服务器消息窗口的输出。如果您在删除或修改数据源时指定 <code>-q</code>，则还必须指定 <code>-y</code>。</p>
-v	<p>将列表的输出格式设置为数行（即表格形式）。</p>
-y	<p>删除或覆盖每个数据源，而不提示您进行确认。如果您在删除或修改数据源时指定 <code>-q</code>，则还必须指定 <code>-y</code>。</p>

细节选项	说明
-c "keyword=value;..."	将连接参数指定为连接字符串。请参见“ 连接参数 ”一节第 248 页。
-cw	确保 DBF 参数（使用 -c 指定）是一个绝对文件名。如果 DBF 的值不是绝对文件名，dbdsn 会在其前面添加当前工作目录 (CWD)。此选项十分有用，因为有些操作系统不会在批处理文件中自动提供 CWD 信息。

注释

修饰符选项可以出现在主选项设定之前或之后。

数据源实用程序是 ODBC 管理器的跨平台替代工具，用于创建、删除、说明和列出 SQL Anywhere ODBC 数据源。此实用程序对于批处理操作是很有用的。

小心

在数据源中存储用户 ID、口令（加密或未加密）和/或数据库密钥是不安全的。如果数据库包含敏感数据，建议您不要将此信息存储在数据源中。

在 Windows 操作系统中，数据源保存在注册表中。

有关使用 ODBC 管理器在 Windows 上创建数据源的信息，请参见“[创建 ODBC 数据源](#)”一节第 90 页。

在 Unix 操作系统中，数据源保存在系统信息文件中（缺省名为 *.odbc.ini*）。在 Unix 中使用数据源实用程序来创建或删除 SQL Anywhere ODBC 数据源时，该实用程序会自动更新系统信息文件的 [ODBC Data Sources] 段。如果在 Unix 中没有使用 -c 选项指定 Driver 连接参数，数据源实用程序会根据 SQLANY11 环境变量的设置自动添加一个带有 SQL Anywhere ODBC 驱动程序完整路径的 Driver 项。

有关系统信息文件的详细信息，请参见“[在 Unix 上使用 ODBC 数据源](#)”一节第 95 页。

小心

在 Unix 中不应使用文件隐藏实用程序 (dbfhide) 对系统信息文件 (*.odbc.ini*) 进行模糊处理，除非您将仅使用 SQL Anywhere 数据源。如果计划使用其它数据源（例如，为了 MobiLink 同步），则系统信息文件的模糊处理可能会妨碍其它驱动程序正常工作。

退出代码是零（成功）或非零（失败）。请参见“[软件组件的退出代码](#)”一节《[SQL Anywhere 服务器 - 编程](#)》。

ODBC 连接参数

数据源实用程序 (dbdsn) 支持下列 ODBC 连接参数。布尔型（true 或 false）参数为 true 时值为 YES 或 1，为 false 时值为 NO 或 0。

名称	说明
Delphi	Delphi 不能处理一行的多个书签值。当您将此值设置为 NO 时，会为每行指派一个书签，而不是像其它情况下那样为每行指派两个书签。将此选项设置为 YES 可提高可滚动游标的性能。
DescribeCursor	<p>此参数用于指定在某过程执行期间重新描述游标的频率。缺省设置为 [根据需要]。</p> <ul style="list-style-type: none"> ● 永不 如果认为不必重新描述游标，则指定 0、N 或 NO。重新描述游标代价较高并且会降低性能。 ● 根据需要 如果想要由 ODBC 驱动程序来确定是否重新描述游标，则指定 1、Y 或 YES。过程中的 RESULT 子句可避免 ODBC 应用程序在游标打开后重新描述结果集。这是缺省设置。 ● 始终 如果指定 2、A 或 ALWAYS，则每次打开游标时都会将其重新描述。如果使用 Transact-SQL 过程或者使用返回多个结果集的过程，则每次打开游标时都必须将其重新描述。
Description	此参数允许您提供 ODBC 数据源的说明。
Driver	<p>此参数用于为连接指定一个 ODBC 驱动程序，方法如下： Driver=driver-name。缺省情况下使用的驱动程序是 SQL Anywhere 11。 <i>driver-name</i> 必须是 SQL Anywhere <i>X</i>，其中 <i>X</i> 是软件的主版本号。<i>driver-name</i> 如果不以 SQL Anywhere 开头，就不能被数据源实用程序 (dbdsn) 读取。</p> <p>在 Unix 中，此参数指定共享对象的完全限定路径。如果在 Unix 中没有指定 Driver 连接参数，数据源实用程序会根据 SQLANY11 环境变量的设置自动添加一个带有 SQL Anywhere ODBC 驱动程序完整路径的 Driver 项。</p>

名称	说明
GetTypeInfoChar	当此选项被设置为 YES 时，会返回 CHAR 列作为 SQL_CHAR（而不是 SQL_VARCHAR）。缺省情况下，会返回 CHAR 列作为 SQL_VARCHAR。
InitString	InitString 允许您指定在建立连接后立即执行的命令。例如，您可能希望设置一个数据库选项或执行一个存储过程。

名称	说明
IsolationLevel	<p>您可以指定下列值之一来设置数据源的初始隔离级别：</p> <ul style="list-style-type: none">● 0 这也称为读取未提交数据隔离级别。这是缺省隔离级别。它提供最大的并发级别，但结果集中会出现脏读、非可重复读取和幻像行。● 1 这也称为读取已提交的数据级别。它提供的并发级别低于级别 0，但消除了级别 0 时结果集中存在的一些不一致。可能会出现非可重复读取和幻像行，但禁止脏读。● 2 这也称为可重复读取级别。可能出现幻像行。禁止脏读和非可重复读取行。● 3 这也称为可序列化级别。这提供最低的并发级别，是最严格的隔离级别。禁止脏读、非可重复读取和幻像行。● 快照 必须为要使用此隔离级别的数据库启用快照隔离。快照隔离级别可以防止读取和写入操作之间的所有干扰。但写入操作之间仍可能会相互干扰。由于争用，可能会出现一些不一致的情况，但性能与隔离级别 0 相同。● 语句快照 必须为要使用此隔离级别的数据库启用快照隔离。快照隔离级别可以防止读取和写入操作之间的所有干扰。但写入操作之间仍可能会相互干扰。由于争用，可能会出现一些不一致的情况，但性能与隔离级别 0 相同。● 只读语句快照 这也称为隔离级别。必须为要使用此隔离级别的数据库启用快照隔离。快照隔离级别可以防止读取和写入操作之间的所有干扰。但写入操作之间仍可能会相互干扰。由于争用，可能会出现一些不一致的情况，但性能与隔离级别 0 相同。

名称	说明
	有关详细信息，请参见“ 选择隔离级别 ”一节《 SQL Anywhere 服务器 - SQL 的用法 》。
KeysInSQLStatistics	如果希望 SQLStatistics 函数返回外键，则指定 YES。ODBC 说明指出 SQLStatistics 不应当返回主键和外键；但是，某些 Microsoft 应用程序（例如 Visual Basic 和 Access）假定主键和外键是由 SQLStatistics 返回的。
LazyAutocommit	将此参数设置为 YES 会延迟提交操作直到语句关闭。
PrefetchOnOpen	当 PrefetchOnOpen 被设置为 YES 时，预取请求随光标打开请求一起发送。预取可使在每次打开光标时，发送读取行的网络请求变得没有必要。列必须已被绑定，以便在打开时进行预取。该连接参数可以帮助减少客户 / 服务器请求数，从而帮助改善 LAN 或 WAN 上的性能。
PreventNotCapable	SQL Anywhere ODBC 驱动程序会返回 [驱动程序不支持] 错误，因为它不支持限定符。某些 ODBC 应用程序不能正确处理该错误。将此参数设置为 YES 可以防止返回此错误代码，从而使这些应用程序能够正常工作。
SuppressWarnings	如果不想显示读取过程中从数据库服务器返回的警告消息，则将此参数设置为 YES。数据库服务器 8.0.0 及更高版本所返回的读取警告比早期版本范围更广。对于用该软件较早版本部署的应用程序，您可以选择此选项以确保正确处理读取警告。
TranslationDLL	提供此选项是为了实现向后兼容。不建议使用转换器。
TranslationName	提供此选项是为了实现向后兼容。不建议使用转换器。
TranslationOption	提供此选项是为了实现向后兼容。不建议使用转换器。

另请参见

- “创建 ODBC 数据源” 一节第 90 页
- “在 Unix 上使用 ODBC 数据源” 一节第 95 页

示例

写入数据源 newdsn 的定义。如果数据源已存在，不提示进行确认。

```
dbdsn -y -w newdsn -c "UID=DBA;PWD=sql;LINKS=TCPIP;ENG=myserver"
```

或者，使用不同的选项顺序，

```
dbdsn -w newdsn -c "UID=DBA;PWD=sql;LINKS=TCPIP;ENG=myserver" -y
```

列出所有已知用户数据源，每行对应一个数据源名称：

```
dbdsn -l
```

列出所有已知系统数据源，每行对应一个数据源名称：

```
dbdsn -ls
```

列出所有数据源及其相关连接字符串：

```
dbdsn -l -b
```

报告用户数据源 MyDSN 的连接字符串：

```
dbdsn -g MyDSN
```

报告系统数据源 MyDSN 的连接字符串：

```
dbdsn -gs MyDSN
```

删除数据源 BadDSN，但首先列出 BadDSN 的连接参数，并提示进行确认：

```
dbdsn -d BadDSN -v
```

删除数据源 BadDSN，但不提示进行确认。

```
dbdsn -d BadDSN -y
```

为数据库服务器 MyServer 创建名为 NewDSN 的数据源：

```
dbdsn -w NewDSN -c "UID=DBA;PWD=sql;ENG=MyServer"
```

如果 NewDSN 已经存在，会提示就是否覆盖该数据源进行确认。

列出所有连接参数名及其别名：

```
dbdsn -cl
```

列出所有用户数据源：

```
dbdsn -l -o dsninfo.txt
```

列出所有连接参数名：

```
dbdsn -cl -o dsninfo.txt
```


指定一个绝对文件名。创建 DSN 时，它将包含 `DBF=c:\SQLAnywhere11\my.db`。

```
c:\SQLAnywhere11> dbdsn -w testdsn -cw -c UID=DBA;PWD=sql;ENG=SQLAny;DBF=my.db
```

生成创建 SQL Anywhere 11 Demo 数据源的命令并将其输出到名为 `restoredsn.bat` 的文件中：

```
dbdsn -cm -gs "SQL Anywhere 11 Demo" > restoredsn.bat
```

`restoredsn.bat` 文件包含以下内容：

```
dbdsn -y -ws "SQL Anywhere 11 Demo" -c "UID=DBA;PWD=sql;  
DBF='C:\Documents and Settings\All Users\Documents\SQL Anywhere 11\Samples  
\demo.db';  
ENG=demo11;START='C:\Program Files\SQL Anywhere 11\bin32\dbeng11.exe';  
ASTOP=yes;Description='SQL Anywhere 11 Sample Database'"
```

返回 Unix 中系统信息文件的位置：

```
dbdsn -f
```

此命令会返回以下输出：

```
dbdsn using /home/user/.odbc.ini
```

更改系统信息文件的位置：

```
export ODBCINI=./myodbc.ini
```

使用 `dbdsn -f` 验证系统信息文件的新位置：

```
dbdsn using ./myodbc.ini
```

创建数据源时使用 `-ns` 选项：

```
dbdsn -w NewDSN -c "UID=DBA" -ns
```

这将产生以下输出：

```
Configuration "newdsn" written to file ./myodbc.ini
```

dbisqlc 实用程序（不建议使用）

dbisqlc 实用程序针对数据库执行 SQL 语句。该实用程序与 Interactive SQL 实用程序 (dbisql) 类似，只不过它不是使用 Java 实现的，如果要将它部署到资源有限的计算机，这一点非常重要。

注意

不建议使用 dbisqlc；但目前不打算将其删除。提供该实用程序的目的是为了在运行 SQL 脚本时保持向后兼容性，并作为一个轻量型部署工具使用。dbisqlc 并不支持 Interactive SQL 所支持的所有功能，可能也不支持当前版本的数据库服务器中可用的所有功能。建议使用 Interactive SQL 实用程序（通过使用 dbisql 命令或选择 [开始] >> [程序] >> [SQL Anywhere 11] >> [Interactive SQL] 进行访问）。请参见“[Interactive SQL 实用程序 \(dbisql\)](#)”一节第 740 页。

语法

dbisqlc [options] [*dbisqlc-command* | *command-file*]

选项	说明
-c " <i>keyword=value; ...</i> "	指定连接参数。如果 Interactive SQL 不能连接，则将显示一个窗口，您可以在其中输入连接参数。请参见“ 连接参数 ”一节第 248 页。
-d <i>delimiter</i>	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。 指定的命令分隔符将被当前 dbisqlc 会话中的所有连接。
-q	取消输出消息。只有在使用命令或命令文件启动 dbrunsql 时，此选项才有意义。指定此选项不会取消错误消息，但会取消以下各项： <ul style="list-style-type: none"> ● 警告和其它非致命错误消息 ● 结果集的输出
-x	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。

注释

dbisqlc 实用程序允许您键入 SQL 命令或运行命令文件。有关 SQL 语句和 Interactive SQL 命令的详细说明，请参见“[SQL 语言元素](#)”《[SQL Anywhere 服务器 - SQL 参考](#)》。

如果指定 *dbisqlc-command*，则 dbisqlc 会执行该命令。您也可以指定命令文件名。如果未指定 *dbisqlc-command* 或 *command-file* 参数，则 dbisqlc 进入交互模式，在这种模式下，您可以在命令窗口中键入命令。

Microsoft Windows、Mac OS X 和 Unix 上支持 dbisqlc 实用程序。

另请参见

- “[Interactive SQL 实用程序 \(dbisql\)](#)”一节第 740 页
- “[SQL 语言元素](#)”《[SQL Anywhere 服务器 - SQL 参考](#)》

示例

以下命令使用用户 ID DBA 和口令 sql，针对当前缺省服务器运行命令文件 *mycom.sql*。如果命令文件中存在错误，则该过程将关闭。

```
dbisqlc -c "UID=DBA;PWD=sql" mycom.sql
```

以下命令会将一个用户添加到当前的缺省数据库中：

```
dbisqlc -c "UID=DBA;PWD=sql" CREATE USER joe IDENTIFIED BY passwd
```

消除实用程序 (dberase)

消除与数据库相关联的 dbspaces 和事务日志文件。

语法

dberase [*options*] *database-file*

选项	说明
@ <i>data</i>	读取来自指定的环境变量或配置文件的选项。请参见“使用配置文件”一节第 695 页。 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“文件隐藏实用程序 (dbfhide)”一节第 724 页。
- ek <i>key</i>	直接在命令行上为高度加密的数据库指定加密密钥。如果您有一个高度加密的数据库，则必须提供加密密钥，才可以使用数据库或事务日志。对于高度加密数据库，必须指定 -ek 或 -ep，但不要同时指定这两者。如果您没有为高度加密数据库指定正确的密钥，则该命令将失败。
- ep	指定需要系统提示输入加密密钥。使用此选项将显示一个窗口，可以在其中输入加密密钥。加密密钥从不以明文显示，因此提供了额外的安全保证。对于高度加密数据库，您必须指定 -ek 或 -ep，但不要同时指定这两者。如果您没有为高度加密数据库指定正确的密钥，则该命令将失败。
- o <i>filename</i>	将输出消息写入指定的文件中。
- q	以安静模式运行—不显示输出消息。如果您指定此选项，则还必须指定 -y，否则操作将失败。
- y	此选项用于删除每个文件，而不提示您进行确认。如果您指定了 -q，则还必须指定 -y，否则操作将失败。

注释

消除实用程序用于消除数据库文件及其关联的事务日志，也可以消除事务日志文件或事务日志镜像文件。所有数据库文件和事务日志文件均已标记为只读，以防止数据库的意外损坏和数据库文件的意外删除。

database-file 可以是数据库文件或事务日志文件。必须指定完整的文件名，包括扩展名。如果指定数据库文件，其关联的事务日志文件会随之消除；如果数据库还维护着一个镜像，该镜像也会随之消除。

注意

消除实用程序不会消除 dbspace。如果要消除 dbspace，可以使用 DROP DATABASE 语句或 Sybase Central 中的 [消除数据库向导]。请参见“[DROP DBSPACE 语句](#)”一节《[SQL Anywhere 服务器 - SQL 参考](#)》。

您还可以使用 [消除数据库向导] 消除 dbspace 和事务日志文件。请参见“[消除数据库](#)”一节第 32 页。

删除引用其它 dbspace 的数据库文件并不会自动删除 dbspace 文件。如果要自己删除 dbspace 文件，请将这些文件从只读更改为可写，然后逐个删除这些文件。或者，也可以使用 DROP DATABASE 语句来消除数据库及其关联的 dbspace 文件。

如果消除数据库文件，其关联的事务日志和事务日志镜像也会随之删除。如果消除一个还维护着事务日志镜像的数据库的事务日志，则该镜像不会随之删除。

使用此实用程序时，要消除的数据库必须处于未运行状态。

退出代码是零（成功）或非零（失败）。请参见“[软件组件的退出代码](#)”一节《[SQL Anywhere 服务器 - 编程](#)》。

文件隐藏实用程序 (dbfhide)

使用简单加密的方式隐藏配置文件和初始化文件的内容。

语法

dbfhide *original-configuration-file encrypted-configuration-file*

选项	说明
<i>original-configuration-file</i>	指定原始文件的名称。
<i>encrypted-configuration-file</i>	指定模糊处理后新文件的名称。

注释

某些实用程序使用配置文件来保存命令行选项。这些选项可能包含口令。可以使用文件隐藏实用程序对配置文件以及 SQL Anywhere 及其实用程序使用的 *.ini* 文件进行简单加密，从而对文件内容进行模糊处理。原来的文件将不会改变。不提供对简单加密过的文件进行解密的功能。要更改模糊处理后的文件，就必须保存原始文件的副本，以便可以修改并再次进行模糊处理。

有关使用配置文件的详细信息，请参见“使用配置文件存储服务器启动选项”一节第 43 页。

有关加密的详细信息，请参见“保护数据的安全”第 1003 页。

隐藏 .ini 文件的内容

通常，SQL Anywhere 要求 *.ini* 文件具有一个特定名称。当您对一个具有重要名称的文件（如 *saldap.ini*）进行简单加密时，需要在对该文件进行简单加密时以不同的名称保存原始文件的副本。如果您没有保存原始文件的副本，那么一旦对其进行了模糊处理，您就不能再修改此文件的内容。以下步骤说明如何对 *.ini* 文件进行简单加密。

◆ 隐藏文件的内容

1. 以不同的名字保存该文件。

```
rename saldap.ini saldap.ini.org
```

2. 使用文件隐藏实用程序对文件进行模糊处理，并将模糊处理后的文件命名为要求的文件名。

```
dbfhide saldap.ini.org saldap.ini
```

3. 利用文件系统或操作系统保护功能来保护 *saldap.ini.org* 文件，或将文件存储到一个安全位置。

要更改 *saldap.ini* 文件，请编辑 *saldap.ini.org* 文件并重复第 2 步。

小心

在 Unix 上，除非仅使用 SQL Anywhere 数据源，否则，不应使用文件隐藏实用程序 (dbfhide) 对系统信息文件（缺省名称是 *.odbc.ini*）进行简单加密。如果计划使用其它数据源（例如，为了 MobiLink 同步），那么对系统信息文件内容进行模糊处理可能会妨碍其它驱动程序正常工作。

此实用程序不支持使用 *@data* 参数从配置文件中读入选项。

另请参见

- [“使用配置文件”一节第 695 页](#)
- [“在配置文件中](#)使用条件分析”一节第 696 页

示例

创建可启动个人数据库服务器和示例数据库的配置文件。应设置 10 MB 的高速缓存，并将个人服务器的此实例命名为 *Elora*。该配置文件编写如下：

```
# Configuration file for server Elora
-n Elora
-c 10M
samples-dir\demo.db
```

(注意，以 # 开头的行是注释行。)

有关 *samples-dir* 的信息，请参见 [“示例目录”一节第 368 页](#)。

将文件命名为 *sample.txt*。如果要用该配置文件启动数据库，命令行应为：

```
dbeng11 @sample.txt
```

现在，对配置文件进行简单加密。

```
dbfhide sample.txt encrypted_sample.txt
```

使用 *encrypted_sample.txt* 文件启动数据库。

```
dbsrv11 @encrypted_sample.txt
```

直方图实用程序 (dbhist)

将直方图转换为包含有关谓词选择性信息的 Microsoft Excel 图表。

语法

dbhist [*options*] -**t** *table-name* [*excel-output-filename*]

选项	说明
@ <i>data</i>	读取来自指定的环境变量或配置文件的选项。请参见“使用配置文件”一节第 695 页。 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“文件隐藏实用程序 (dbfhide)”一节第 724 页。
- c <i>options</i>	指定连接参数。请参见“连接参数”一节第 248 页。
- n <i>colname</i>	指定要与直方图关联的列的名称。如果不指定列，则返回表中有直方图的所有列。
- t <i>table-name</i>	指定要生成图表的表或实例化视图的名称。
- u <i>owner</i>	指定表或实例化视图的所有者。
<i>excel-output-name</i>	指定所生成 Excel 文件的名称。如果未指定名称，则 Excel 提示您在 [另存为] 窗口中输入一个名称。

注释

直方图存储在 ISYSCOLSTAT 系统表中，也可以使用 sa_get_histogram 存储过程进行检索。直方图实用程序将直方图转换为 Microsoft Excel 图表，图表包含有谓词选择性的信息。直方图实用程序 (dbhist) 只能在 Windows 上运行，而且必须安装 Excel 97 或更高版本。

表或实例化视图可能不会显示统计信息（包括直方图），例如，对于最近已删除统计信息的情况。此时，直方图实用程序会返回消息 [直方图中没有任何数据，正在中止]。在这种情况下，必须创建统计信息，然后再次运行直方图实用程序。要为表或实例化视图创建统计信息，需执行 CREATE STATISTICS 语句。请参见“CREATE STATISTICS 语句”一节《SQL Anywhere 服务器 - SQL 参考》。

要确定关于字符串的谓词选择性，应该使用 ESTIMATE 或 ESTIMATE_SOURCE 函数。尝试从字符串中检索直方图会导致 sa_get_histogram 和直方图实用程序产生错误。请参见“ESTIMATE 函数 [Miscellaneous]”一节《SQL Anywhere 服务器 - SQL 参考》和“ESTIMATE_SOURCE 函数 [杂类]”一节《SQL Anywhere 服务器 - SQL 参考》。

工作表以列名来命名。列名达到 24 个字符后会被截断，所有出现的 \、/、?、*、[、] 和 :（Excel 中不允许使用这些字符）均被替换为下划线 ()。图表名以 chart 一词为前缀，后面部分的命名约定与上面相同。重复的名称（因字符替换、截断或名称以 chart 开头的列引起）会引发 Excel 错误，错误消息会提示不能使用重复名称。但电子表格仍会以之前创建版本的名称（Sheet1、Chart1 等等）来创建。

退出代码是零（成功）或非零（失败）。请参见“软件组件的退出代码”一节《SQL Anywhere 服务器 - 编程》。

您还可以使用 `sa_get_histogram` 存储过程检索直方图。请参见“`sa_get_histogram` 系统过程”一节《SQL Anywhere 服务器 - SQL 参考》。

示例

以下命令（全部在同一行输入）将为数据库 `demo.db` 的表 `SalesOrderItems` 中的 `ProductID` 列生成 Excel 图表，并将其另存为 `histogram.xls`。

```
dbhist -c "UID=DBA;PWD=sql;DBF=samples-dir\demo.db" -n ProductID -t
SalesOrderItems histogram.xls
```

假定示例数据库已经启动，以下语句会为表 `SalesOrders` 中有直方图的每个列生成图表。此语句会尝试使用 `UID=DBA` 和 `PWD=sql` 进行连接。由于未指定输出文件名，因此 Excel 会提示您输入文件名。

```
dbhist -t SalesOrders -c "UID=DBA;PWD=sql"
```

有关 `samples-dir` 的信息，请参见“示例目录”一节第 368 页。

另请参见

- “优化程序估计值和列统计信息”一节《SQL Anywhere 服务器 - SQL 的用法》
- “CREATE STATISTICS 语句”一节《SQL Anywhere 服务器 - SQL 参考》

信息实用程序 (dbinfo)

显示有关指定数据库的信息。

语法

dbinfo [*options*]

选项	说明
@ <i>data</i>	读取来自指定的环境变量或配置文件的选项。请参见“使用配置文件”一节第 695 页。 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“文件隐藏实用程序 (dbfhide)”一节第 724 页。
-c " <i>keyword=value; ...</i> "	指定连接参数。请参见“连接参数”一节第 248 页。 任何有效的用户 ID 都可以运行信息实用程序；但是要获得页面使用统计信息，则需要具有 DBA 权限。
-o <i>filename</i>	将输出消息写入指定的文件中。
-q	以安静模式运行—不显示消息。
-u	显示有关所有表（其中包括系统表 and 用户定义的表）以及实例化视图的使用情况和大小的信息。 只有在没有其他用户连接到数据库并且您拥有 DBA 权限时，您才可以请求页面使用统计信息。页面使用信息通过 sa_table_page_usage 系统过程获取。

注释

dbinfo 实用程序用于显示数据库的相关信息。它可以报告数据库文件的名称、任何事务日志文件或日志镜像的名称、页面大小、归类名称和标签、是否启用了表加密，以及其它信息。另外，它还可以提供有关表使用情况的统计数据 and 详细信息。

可使用 dbinfo 实用程序来确定磁盘上某个表的大小。为此，请运行如下所示的命令：

```
dbinfo -u -c "UID=DBA;PWD=sql;DBF=sample-dir\demo.db"
```

结果将向您展示有多少页面用于保存数据库中每个表内的数据 (Pages)，以及这些页面的使用百分比 (%used)。对于任意表，随后可用页数乘以数据库页面大小，然后再用该值乘以 %used 来确定该表所占用的空间量。

退出代码是零 (成功) 或非零 (失败)。

有关退出代码的详细信息，请参见“软件组件的退出代码”一节《SQL Anywhere 服务器 - 编程》。

初始化实用程序 (dbinit)

创建新数据库。

语法

dbinit [*options*] *new-database-file*

选项	说明
@data	<p>读取来自指定的环境变量或配置文件的选项。请参见“使用配置文件”一节第 695 页。</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“文件隐藏实用程序 (dbfhide)”一节第 724 页。</p>
-a	<p>如果对 CHAR 或 NCHAR 数据类型使用 UCA (Unicode 归类算法) (请参见 -z 和 -zn)，使用该选项可以在进行字符串比较时考虑字母间重音的差异 (例如，e 小于 é)。除使用 UCA 归类创建的日文数据库外，缺省情况下会忽略重音差异 (也就是说 e 等于 é)。如果所有基本字母 (去除了重音和大小写的字母) 均相等，则从左向右来比较重音。</p> <p>创建日文数据库时 UCA 归类的区分重音缺省值为区分。也就是说，会考虑重音。请参见“Unicode 归类算法 (UCA)”一节第 393 页。</p>
-af	<p>如果对 CHAR 或 NCHAR 数据类型使用 UCA (请参见下文的 -z 和 -zn)，使用该选项可以在进行字符串比较时考虑字母间重音的差异 (例如，e 小于 é)。缺省情况下会忽略重音差异 (也就是说 e 等于 é)。如果所有基本字母 (去除了重音的字母) 均相等，则按照法语规则从右向左来比较重音。</p> <p>有关详细信息，请参见“Unicode 归类算法 (UCA)”一节第 393 页。</p>

选项	说明
-b	<p>用空白填充数据库。</p> <p>SQL Anywhere 在比较所有字符串时都会将它们视为具有可变的长度并存储在 VARCHAR 域。这包括涉及固定长度 CHAR 或 NCHAR 列的字符串比较。此外，当数值存储于数据库中时，SQL Anywhere 不会截断或添加尾随空白。</p> <p>缺省情况下，SQL Anywhere 将空白视为有效字符。因此，值 'a'（后跟一个空格的字符 'a'）不等同于单个字符串 'a'。不等比较还会将空白与归类中的任何其它字符同等看待。</p> <p>如果启用空格填充（dbinit -b 选项），则字符串比较的语义更符合 ANSI/ISO SQL 标准。启用空格填充后，SQL Anywhere 会在所有比较中忽略尾随空白。</p> <p>在上例中，在采用空格填充的数据库中 'a' 与 'a' 的相等比较会返回 TRUE。对于采用空格填充的数据库，固定长度的字符串值在由应用程序读取时会以空白进行填充。ansi_blanks 连接选项可控制应用程序针对此类赋值是否会收到字符串截断警告。请参见“ansi_blanks 选项 [兼容性]”一节第 480 页。</p>
-c	<p>在比较操作和字符串操作中将所有值都视为是区分大小写的。数据库中的标识符不区分大小写，即使在区分大小写的数据库中也是如此。</p> <p>除使用 UCA 归类创建的日文数据库外，其缺省行为是所有比较都不区分大小写。创建日文数据库时 UCA 归类的区分大小写的缺省值为区分大小写。</p> <p>用作 QAnywhere 服务器存储库的数据库应区分大小写。</p> <p>提供此选项是为了与 ISO/ANSI SQL 标准兼容。</p>

选项	说明
-dba [<i>DBA-user</i>][<i>,pwd</i>]	<p>指定 DBA 用户 ID 和口令。如果为数据库 DBA 用户指定新的名称，即无法再以用户 DBA 身份连接到数据库。您也可以为 DBA 数据库用户指定一个不同的口令。如果不指定口令，则使用缺省口令 sql。如果不指定此选项，则将以口令 sql 创建缺省的用户 ID DBA。</p> <p>下面两条命令都将创建一个数据库，其 DBA 用户名为 testuser，使用缺省口令 sql:</p> <pre>dbinit -dba testuser mydb.db dbinit -dba testuser, mydb.db</pre> <p>以下命令使用缺省用户 ID DBA 和口令 mypwd:</p> <pre>dbinit -dba ,mypwd mydb.db</pre> <p>以下命令将 DBA 用户更改为 user1，口令为 mypwd:</p> <pre>dbinit -dba user1,mypwd mydb.db</pre> <p>建议采用由 7 位 ASCII 字符组成的口令，因为如果服务器不能将客户端的字符集转换为 UTF-8，其它字符可能会无法正常工作。</p>
-dbs size [k m g p]	<p>为数据库预分配空间。预先为数据库分配空间有助于降低数据库所在驱动器空间不足的风险。而且，它还可以在数据库服务器有数据库扩充需要之前增加数据库的可存储数据量，从而有助于提高各项性能。扩充数据库是一项非常耗时的操作。</p> <p>缺省情况下，<i>size</i> 以字节为单位。可以分别使用 k、m、g 或 p 将单位指定为千字节、兆字节、千兆字节或页数。</p>

选项	说明
<p>-ea algorithm</p>	<p>指定用于数据库或表加密 (-et) 的加密算法。指定 -ea simple 可进行简单加密（不要指定 -ek 或 -ep）。简单加密等效于模糊处理，其目的只是为了使数据处于隐藏状态，以防有人意外地直接访问数据库文件，从而使要破译数据库中数据的人难于使用磁盘实用程序查看文件。</p> <p>为获得更高的安全性，可指定 AES 或 AES256 以分别实现 128 位或 256 位高度加密。指定 AES_FIPS 或 AES256_FIPS 以分别实现 128 位或 256 位 FIPS 认可的高度加密。对于高度加密，还必须指定 -ek 或 -ep 选项。有关高度加密的详细信息，请参见“高度加密”一节第 1019 页。</p> <p>要创建未加密的数据库，请指定 -ea none 或者不包含 -ea 选项（并且不要指定 -et、-ep 或 -ek）。</p> <p>如果未指定 -ea 选项，则缺省行为如下：</p> <ul style="list-style-type: none"> ● 如果未指定 -ek、-ep 或 -et，则为 -ea none ● 如果指定了 -ek 或 -ep（带或不带 -et），则为 -ea AES ● 如果指定了 -et，而未指定 -ek 或 -ep，则为 -ea simple <p>算法名称是不区分大小写的。</p> <p>在 Windows Mobile 上，仅 ARM 处理器支持 AES_FIPS 和 AES256_FIPS 算法。</p> <p>以下命令会创建一个高度加密的数据库，并指定加密密钥和算法。</p> <pre>dbinit -ek "0kZ2o56AK#" -ea AES_FIPS "myencrypteddb.db"</pre> <p>文件压缩实用程序在压缩加密数据库文件时，无法达到未加密数据库文件那样的压缩程度。</p> <div style="border: 1px solid black; padding: 5px;"> <p>需要单独授予许可的组成部分</p> <p>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。</p> <p>请参见“单独授权的组件”一节《SQL Anywhere 11 - 简介》。</p> </div>

选项	说明
-ek <i>key</i>	<p>通过直接在命令中指定加密密钥，来指定您希望创建高度加密的数据库。-ek 选项和 AES 算法一起使用，可选择使用 -ea 选项来指定该算法。如果您指定了 -ek 选项而没有指定 -ea 选项，则缺省情况下使用 AES 算法。</p> <p>如果同时也指定 -et，则不会将数据库加密，而是启用表加密。请参见“表加密”一节第 1026 页。</p> <div style="border: 1px solid black; padding: 5px; margin-top: 10px;"> <p>小心</p> <p>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。</p> </div>
-ep	<p>指定您希望通过在窗口中输入加密密钥来创建高度加密的数据库。这样，加密密钥决不会以明文显示，从而提供了额外的安全保证。</p> <p>加密密钥必须输入两次，以确认密钥的输入是正确的。如果密钥不匹配，则初始化失败。</p> <p>如果同时也指定 -et，则不会将数据库加密，而是启用表加密。</p> <p>有关详细信息，请参见“高度加密”一节第 1019 页。</p>
-et	<p>使用为 -ea 选项指定的加密算法（和密钥）启用表加密。此选项在需要创建加密表时使用，而不是在加密整个数据库时使用。如果同时指定 -et 与 -ek 或 -ep 而不是 -ea，将缺省采用 AES 算法。如果仅指定 -et，则采用简单加密。</p> <p>启用表加密并不表示您的表就会被加密。您必须在创建数据库后分别加密每张表。请参见“加密表”一节第 1027 页。</p> <p>启用表加密时，将对加密表的表页、相关联的索引页和临时文件页加密，并对包含有关加密表的事务的事务日志页加密。</p> <p>以下示例会创建为表启用了高度加密的数据库 <i>new.db</i>，该加密使用密钥 <i>abc</i> 和 AES_FIPS 加密算法：</p> <pre style="color: red; font-family: monospace;">dbinit -et -ek abc -ea AES_FIPS new.db</pre>

选项	说明
-i	<p>从数据库中排除 jConnect 系统对象。如果要使用 jConnect JDBC 驱动程序来访问系统目录信息，则需要安装 jConnect 目录支持（缺省情况下会安装）。指定此选项后，只要不访问系统信息，就仍然可以使用 JDBC。以后需要 jConnect 支持时，可以使用 Sybase Central 或 ALTER DATABASE 语句来添加它。</p> <p>有关详细信息，请参见“在数据库中安装 jConnect 系统对象”一节《SQL Anywhere 服务器 - 编程》。</p> <p>如果要创建在 Windows Mobile 上使用的数据库，请参见“在 Windows Mobile 上使用 jConnect”一节第 322 页。</p>
-k	<p>不创建 SYSCOLUMNS 视图和 SYSINDEXES 视图。缺省情况下，在创建数据库时会生成视图 SYS.SYSCOLUMNS 和 SYS.SYSINDEXES，以便与 Watcom SQL（本软件的版本 4 及更低版本）中所提供的系统表兼容。这些视图与 Sybase Adaptive Server Enterprise 兼容性视图 dbo.syscolumns 和 dbo.sysindexes 冲突。</p>
-l	<p>列出推荐的归类序列，然后停止。不创建数据库。在 Sybase Central 的 [创建数据库向导] 中，自动提供可用归类序列的列表。</p>
-le	<p>列出可用字符集编码，然后停止。不创建数据库。每个字符集编码由一个或多个标签标识。这些是可用于标识编码的字符串。所显示的每一行文本会列出用于标识编码的编码标签和替代标签。这些标签分为几种常用类别：SA（SQL Anywhere 标签）、IANA（Internet 编号授权委员会）、MIME（多用途 Internet 邮件扩展）、ICU（Unicode 的国际组件）、JAVA 或 ASE (Adaptive Server Enterprise)。</p> <p>如果想要查看包含替换标签的字符集编码列表，请指定 -le+ 选项。</p> <p>当初始化实用程序报告字符集编码时，它始终报告 SQL Anywhere 版本的标签。例如，以下命令会报告 CHAR 字符集编码 windows-1250:</p> <pre>dbinit -ze cp1250 -z uca test.db</pre>

选项	说明
-m filename	创建一个事务日志镜像。事务日志镜像是与事务日志完全相同的一个副本，通常在单独的设备上进行维护，以便更好地保护您的数据。缺省情况下，SQL Anywhere 不使用事务日志镜像。
-n	创建一个没有事务日志的数据库。创建没有事务日志的数据库可节省磁盘空间，但将导致性能较差，因为每次提交都会执行检查点操作。而且，如果数据库损坏并且运行时没有事务日志，数据将无法恢复。事务日志是数据复制所必需的，可在出现介质故障或系统故障时为数据库信息提供额外的安全保护。
-o filename	将输出消息写入指定的文件中。
-p page-size	<p>指定数据库的页面大小。数据库的页面大小可以是（以字节为单位）2048、4096、8192、16384 或 32768，缺省值是 4096。</p> <p>较大的页面大小对于大型数据库而言更有益。例如，由于一次读取一整页，因此扫描表所需的 I/O 操作数通常较少。但是，较大的页面大小会导致额外的内存要求。强烈建议在选择页面大小时进行性能测试（和一般测试）。然后选择可以产生满意结果的最小的页面大小。对于大多数应用程序，建议不要使用 16 KB 或 32 KB 页面大小。只有确定始终可以使用大容量的数据库服务器高速缓存，而且，只有在对内存和磁盘空间与其性能特点进行了权衡之后，才可以在生产系统中使用 16 KB 或 32 KB 的页面大小。如果要在同一服务器上启动大量数据库，请选择一个合理的页面大小。</p> <p>有关详细信息，请参见：</p> <ul style="list-style-type: none"> ● “使用适当的页大小”一节 《SQL Anywhere 服务器 - SQL 的用法》 ● “表大小和页面大小”一节 《SQL Anywhere 服务器 - SQL 的用法》
-q	以安静模式运行—不显示消息。

选项	说明
-s	<p>向数据库页添加校验和。校验和用于确定是否在磁盘上已修改了数据库页。在您创建启用校验和的数据库时，在将每一页写入磁盘前为其计算校验和。下次从磁盘中读取该页时，就会重新计算该页的校验和，并将其与该页上存储的校验和进行比较。如果校验和不同，说明页面在磁盘上已被修改或损坏，因此会产生一个错误。无论是否指定 -s，数据库服务器都会对重要的数据库页面进行校验和检查。</p> <p>系统会自动为在 Windows Mobile 以及存储设备（例如可移动驱动器）上运行的数据库启用校验和，以帮助在数据库损坏时尽早地将其检测出来。</p>
-t <i>transaction-log-name</i>	<p>指定事务日志文件的名称。事务日志是数据库服务器用来记录所有用户所做的所有更改的文件，无论使用哪个应用程序。在备份和恢复（请参见“事务日志”一节第 14 页）以及数据复制中，事务日志起着重要的作用。如果文件名没有路径，它将被放在与数据库文件相同的目录中。如果运行 dbinit 但不指定 -t 或 -n，则会创建与数据库文件同名的事务日志，但其扩展名为 <i>.log</i>。</p>

选项	说明
<p>-z coll [<i>collation-tailoring-string</i>]</p>	<p>指定数据库的归类序列。归类序列用于字符数据类型 (CHAR、VARCHAR 和 LONG VARCHAR) 的排序和比较。归类为所使用的编码 (字符集) 提供字符比较和排序信息。谨慎选择归类非常重要。一旦数据库创建完毕, 如果不卸载并重装数据库, 就无法更改归类。如果不指定归类, SQL Anywhere 会根据操作系统语言和字符集选择一种归类。请参见:</p> <ul style="list-style-type: none"> ● “选择归类” 一节第 395 页 ● “建议的字符集和归类” 一节第 408 页 ● “支持的归类和替代归类” 一节第 404 页 <p>您还可以指定归类定制选项 (<i>collation-tailoring-string</i>), 以便能够对字符的排序和比较进行更多的控制。这些选项采用 "关键字=值" 对的形式, 包括在圆括号内, 位于归类名后。例如:</p> <pre>dbinit -c -z uca(locale=es;case=LowerFirst) spanish2.db</pre> <p>请参见 “归类定制选项” 一节第 396 页。</p> <p>若指定了这两者, 则 <i>collation-tailoring-string</i> 中指定的大小写和重音设置会替换 dbinit (-c、-a 和 -af) 的大小写和重音选项。</p> <div style="border: 1px solid black; padding: 5px; margin-top: 10px;"> <p>注意 通过归类定制选项初始化的数据库无法通过 10.0.1 之前版本的数据库服务器启动。</p> </div>
<p>-ze encoding</p>	<p>为归类指定编码。通过 -z 所指定的大多数归类都会规定出编码 (字符集) 和排序信息。对于这些归类, 不应指定 -ze。</p> <p>如果 -z 所指定的归类为 UCA (Unicode 归类算法), 则 -ze 可以针对 CHAR 数据类型指定 UTF-8 或任何单字节编码。缺省情况下, SQL Anywhere 会使用 UTF-8。使用 -ze 可以指定特定于区域设置的编码并利用 UCA 进行比较和排序。</p>

选项	说明
<p>-zn coll [<i>collation-tailoring-string</i>]</p>	<p>指定用于排序和比较国家字符数据类型（NCHAR、NVARCHAR 和 LONG NVARCHAR）的归类序列。该归类为所用的 UTF-8 编码（字符集）提供字符排序信息。其值为 UCA（缺省值）或 UTF8BIN（提供编码大于 0x7E 的所有字符的二进制排序）。如果未安装 dbicu11 和 dbicudt11 DLL，那么缺省 NCHAR 归类为 UTF8BIN。有关详细信息，请参见“选择归类”一节第 395 页。</p> <p>您还可以指定归类定制选项 (<i>collation-tailoring-string</i>)，以便能够对字符的排序和比较进行更多的控制。这些选项采用 "关键字=值" 对的形式，包括在圆括号内，位于归类名后。例如：</p> <pre>dbinit -c -zn UCA(case=LowerFirst) sens.db</pre> <p>请参见“归类定制选项”一节第 396 页。</p> <p>若指定了这两者，则 <i>collation-tailoring-string</i> 中指定的大小写和重音设置会替换 dbinit (-c、-a 和 -af) 的大小写和重音选项。</p> <div style="border: 1px solid black; padding: 5px; margin-top: 10px;"> <p>注意 通过归类定制选项初始化的数据库无法通过 10.0.1 之前版本的数据库服务器启动。</p> </div>

注释

有若干个数据库属性在初始化时指定。除非卸载、重新初始化和重建整个数据库，否则以后无法更改这些属性。这些数据库属性包括：

- 区分大小写或不区分大小写
- 区分重音或不区分重音
- 区分标点符号
- 比较时尾随空白的处理方式
- 页面大小
- 字符集编码或归类序列
- 数据库加密
- 表加密

例如，可按如下所示创建页面大小为 8192 字节的数据库 *test.db*：

```
dbinit -p 8192 test.db
```

您不能将数据库命名为 **utility_db**。该名称已保留供实用程序数据库使用。请参见“[使用实用程序数据库](#)”一节第 28 页。

当在初始化命令中指定归类定制选项时，如果数据库不区分大小写或重音，将无法为标点符号区分指定四进制。

另外，在进行初始化时选择是否使用事务日志和事务日志镜像。此选择可以在以后使用事务日志实用程序或 ALTER DATABASE 语句进行更改。

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》。

还可以通过以下方法来创建数据库：

- 从 Sybase Central 使用 [创建数据库向导]。请参见“创建数据库 (Sybase Central)”一节第 20 页。
- 从 Interactive SQL 使用 CREATE DATABASE 语句。请参见“CREATE DATABASE 语句”一节《SQL Anywhere 服务器 - SQL 参考》。

注意

当部署应用程序时，需要使用个人数据库服务器 (dbeng11)，通过 dbinit 实用程序来创建数据库。如果在没有运行其它数据库服务器时，从本地计算机的 Sybase Central 创建数据库，也需要使用个人数据库服务器。

退出代码是零（成功）或非零（失败）。请参见“软件组件的退出代码”一节《SQL Anywhere 服务器 - 编程》。

示例

以下命令创建区分大小写的数据库 *spanish.db*，它对非 NCHAR 数据使用 1262spa 归类。对于 NCHAR 数据，指定 UCA 归类，地区设置为 es，并且按小写字母在前排序。

```
dbinit -c -z 1252spa -zn uca(locale=es;case=LowerFirst) spanish.db
```

Interactive SQL 实用程序 (dbisql)

针对数据库执行 SQL 命令并运行命令文件。

语法

dbisql [*options*] [*dbisql-command* | *command-file*]

选项	说明
@data	<p>读取来自指定的环境变量或配置文件的选项。请参见“使用配置文件”一节第 695 页。</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“文件隐藏实用程序 (dbfhide)”一节第 724 页。</p>
-c " <i>keyword=value; ...</i> "	<p>指定连接参数。如果 Interactive SQL 不能连接，则将显示一个窗口，您可以在其中输入连接参数。请参见“连接参数”一节第 248 页。</p>
-d delimiter	<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>此选项会替换 <code>command_delimiter</code> 选项的设置。请参见“<code>command_delimiter</code> 选项 [Interactive SQL]”一节第 670 页。</p>
-dl	<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。（最后一个字符是数字 1，而不是小写的 L）。只有在将 Interactive SQL 作为命令行程序运行时，才可以使用此选项。</p>
-datasource DSN-name	<p>指定要连接的 ODBC 数据源。</p>
-f filename	<p>在 [SQL 语句] 窗格中打开（但不运行）名为 <i>filename</i> 的文件。</p> <p>如果有 -f 选项，将忽略 -c 选项；即不会连接数据库。</p> <p>文件名可用引号括起来；如果文件名包含空格，则必须用引号括起来。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，然后退出。如果文件名不包括完整的驱动器和路径说明，则假定它相对于当前目录。</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>
-host hostname	<p>指定正运行数据库服务器的计算机的 <i>hostname</i> 或 IP 地址。可以使用名称 <code>localhost</code> 来代表当前计算机。</p>

选项	说明
-nogui	<p>在命令提示符模式下运行 Interactive SQL，不使用窗口式用户界面。这对于批处理操作是很有用的。如果指定 <i>dbisql-command</i> 或 <i>command-file</i>，则假定已指定 -nogui。</p> <p>在此模式下，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，可使用环境变量 ERRORLEVEL 检查程序退出代码。请参见“软件组件的退出代码”一节《SQL Anywhere 服务器 - 编程》。</p>
-onerror { continue exit }	<p>控制在从命令文件读取语句时如果遇到错误将采取的操作。此选项会替换 on_error 设置。在批处理操作中使用 Interactive SQL 时，它是很有用的。请参见“on_error 选项 [Interactive SQL]”一节第 679 页。</p>
-port port-number	<p>指定数据库服务器使用的端口号。SQL Anywhere 的缺省端口号是 2638。</p>
-q	<p>取消输出消息。只有在使用命令或命令文件启动 Interactive SQL 时，此选项才有意义。指定此选项不会取消错误消息，但会取消以下各项：</p> <ul style="list-style-type: none"> ● 警告和其它非致命错误消息 ● 结果集的输出
-ul	<p>指定 UltraLite 数据库为缺省数据库。Interactive SQL 会根据您连接到的数据库类型来决定可供使用的选项。缺省情况下，Interactive SQL 会假定您连接的是 SQL Anywhere 数据库。如果指定 -ul 选项，缺省设置会更改为 UltraLite 数据库。不管已将哪种数据库设置为缺省数据库，您都可以通过在 [连接] 窗口的下拉列表中选择需要的数据库类型来连接到 SQL Anywhere 或 UltraLite 数据库。</p> <p>有关从 Interactive SQL 连接到 UltraLite 数据库的详细信息，请参见“UltraLite 的 Interactive SQL 实用程序 (dbisql)”一节《UltraLite - 数据库管理和参考》。</p>
-version	<p>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]。</p>
-x	<p>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。</p> <p>有关 SQL 语句和 Interactive SQL 命令的详细说明，请参见“SQL 语言元素”《SQL Anywhere 服务器 - SQL 参考》。</p>

注释

Interactive SQL 允许您浏览数据库，执行 SQL 命令，并运行命令文件。它还提供有关受影响的行数、每个命令所需的时间、查询的执行计划以及任何错误消息的反馈。

您可以连接到 SQL Anywhere 和 UltraLite 数据库。

Interactive SQL 可在 Windows、Solaris、Linux 和 Mac OS X 上使用。

如果指定 *dbisql-command*，则 Interactive SQL 会执行该命令。您也可以指定命令文件名。如果未指定 *dbisql-command* 或 *command-file* 参数，则 Interactive SQL 进入交互模式，在这种模式下，您可以在命令窗口中键入命令。

您可以通过下列方法来启动 Interactive SQL：

- 从 Sybase Central 使用 [打开 Interactive SQL] 菜单项。
- 在 [开始] 菜单中选择 [开始] » [程序] » [SQL Anywhere 11] » [Interactive SQL]。
- 使用 *dbisql* 命令。

对于 Windows，有两个可执行文件。批处理脚本应调用 *dbisql* 或 *dbisql.com* 而不是 *dbisql.exe*。*dbisql.com* 可执行文件作为控制台应用程序进行链接。*dbisql.exe* 可执行文件作为窗口式应用程序进行链接，且不止用于启动该应用程序的命令 shell。如果 *dbisql.exe* 从批处理文件运行，您将不会看到有任何输出发送到标准输出或标准错误文件。

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 Windows XP 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;
```

还可使用 `default_isql_encoding` 选项来设置 Interactive SQL 的缺省代码页。请参见：

- “建议的字符集和归类”一节第 408 页
- “`default_isql_encoding` 选项 [Interactive SQL]”一节第 671 页
- “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》

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。请参见“软件组件的退出代码”一节 《SQL Anywhere 服务器 - 编程》。

通过 Interactive SQL 执行 *reload.sql* 文件时，必须将加密密钥指定为一个参数。如果未在 **READ** 语句中提供密钥，Interactive SQL 会提示您输入密钥。

另请参见

- “CLEAR 语句 [Interactive SQL]” 一节 《SQL Anywhere 服务器 - SQL 参考》
- “CONFIGURE 语句 [Interactive SQL]” 一节 《SQL Anywhere 服务器 - SQL 参考》
- “CONNECT 语句 [ESQL] [Interactive SQL]” 一节 《SQL Anywhere 服务器 - SQL 参考》
- “DESCRIBE 语句 [Interactive SQL]” 一节 《SQL Anywhere 服务器 - SQL 参考》
- “EXIT 语句 [Interactive SQL]” 一节 《SQL Anywhere 服务器 - SQL 参考》
- “HELP 语句 [Interactive SQL]” 一节 《SQL Anywhere 服务器 - SQL 参考》
- “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》
- “PARAMETERS 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》
- “SET CONNECTION 语句 [Interactive SQL] [ESQL]” 一节 《SQL Anywhere 服务器 - SQL 参考》
- “SET OPTION 语句 [Interactive SQL]” 一节 《SQL Anywhere 服务器 - SQL 参考》
- “START ENGINE 语句 [Interactive SQL]” 一节 《SQL Anywhere 服务器 - SQL 参考》
- “START LOGGING 语句 [Interactive SQL]” 一节 《SQL Anywhere 服务器 - SQL 参考》
- “STOP LOGGING 语句 [Interactive SQL]” 一节 《SQL Anywhere 服务器 - SQL 参考》
- “SYSTEM 语句 [Interactive SQL]” 一节 《SQL Anywhere 服务器 - SQL 参考》

示例

以下命令使用用户 ID DBA 和口令 sql，针对当前缺省服务器运行命令文件 *mycom.sql*。如果命令文件中存在错误，则该过程将关闭。

```
dbisql -c "UID=DBA;PWD=sql" -onerror exit mycom.sql
```

以下命令会将一个用户添加到当前的缺省数据库中：

```
dbisql -c "UID=DBA;PWD=sql" CREATE USER joe IDENTIFIED passwd
```

密钥对生成器实用程序 (createkey)

创建 RSA 和 ECC 密钥对，用于实现 MobiLink 端对端加密。

语法

createkey

注释

要创建 ECC 对象，您必须订购一个单独的许可。请参见“[单独授权的组件](#)”一节《[SQL Anywhere 11 - 简介](#)》。

运行 createkey 时，系统会提示您输入以下信息：

- **选择加密类型** 仅当购买了用于 ECC 加密的许可时，此提示才会显示。选择 RSA 或 ECC。
- **输入 RSA 密钥长度 (512-16384)** 只有选择了 RSA 加密，此提示才会显示。可以选择一个介于 512 位到 16384 位之间的长度。
- **输入 ECC 曲线** 只有在已购买了用于 ECC 加密的许可而且选择了 ECC 加密类型时，此提示才会显示。系统会提示您从 ECC 曲线列表中进行选择。缺省值为 sect163k1。
- **输入文件路径以保存公共密钥** 为生成的 PEM 编码的公共密钥指定文件名和位置。在 MobiLink 客户端通过 e2ee_public_key 协议选项指定此文件。请参见“[e2ee_public_key](#)”一节《[MobiLink - 客户端管理](#)》。
- **输入文件路径以保存专用密钥** 为生成的 PEM 编码的专用密钥指定文件名和位置。在 MobiLink 服务器上通过 e2ee_private_key 协议选项指定此文件。请参见“[-x 选项](#)”一节《[MobiLink - 服务器管理](#)》。
- **输入口令以保护专用密钥** 或者提供用来加密专用密钥的口令。如果不提供口令，专用密钥就不会被加密。在 MobiLink 服务器上通过 e2ee_private_key_password 协议选项指定此口令。请参见“[-x 选项](#)”一节《[MobiLink - 服务器管理](#)》。

另请参见

- “[端对端加密](#)”一节第 1047 页
- “[e2ee_type](#)”一节《[MobiLink - 客户端管理](#)》（MobiLink 客户端网络协议选项）

示例

以下示例将创建一个 RSA 密钥对：

```
>createkey
SQL Anywhere 密钥对生成器版本 11.0.0.1304
选择加密类型 ((R)SA 或 (E)CC): r
输入 RSA 密钥长度 (512-16384): 2048
正在生成密钥对...
输入文件路径以保存公共密钥: rsapublic.pem
输入文件路径以保存专用密钥: rsaprivate.pem
输入口令以保护专用密钥: pwd
```

语言选择实用程序 (dblang)

报告和更改那些用于控制 SQL Anywhere 和 Sybase Central 使用的语言的注册表设置。

语法

dblang [*options*] *language-code*

选项	说明
-m	将语言代码写入注册表中的 HKEY_LOCAL_MACHINE 下。
-q	以安静模式运行—不输出消息。
-u	将语言代码写入注册表中的 HKEY_CURRENT_USER 下。这是缺省位置。

语言代码	语言
EN	英语
DE	德语
ES	西班牙语
FR	法语
IT	意大利语
JA	日语
KO	朝鲜语
LT	立陶宛语
PL	波兰语
PT	葡萄牙语
RU	俄语
TW	繁体中文
UK	乌克兰语
ZH	简体中文

注释

如果未指定 **-m** 或 **-u**，则会将语言代码写入注册表中的 HKEY_CURRENT_USER 下。可以同时指定 **-m** 和 **-u**，将语言代码写入上述两个位置。

如果运行 `dblang` 实用程序时不使用语言代码，则报告当前设置。这些设置如下：

- **SQL Anywhere** 此设置控制着使用哪个语言资源库来生成来自 SQL Anywhere 数据库服务器的信息性消息和错误消息。语言资源库是一个 DLL，其名称采用 `dblgXX11.dll` 形式，其中 `XX` 是双字母语言代码。

请确保在更改这些设置时，您的计算机上具有相应的语言资源库。

- **Sybase Central** 此设置控制用于显示 Sybase Central 和 Interactive SQL 的用户界面元素的资源。您必须购买 SQL Anywhere 相应的本地化版本，才能使此设置生效。

退出代码是零（成功）或非零（失败）。请参见“[软件组件的退出代码](#)”一节《[SQL Anywhere 服务器 - 编程](#)》。

此实用程序不支持使用 `@data` 参数从配置文件中读入选项。

如果启用了快速启动程序选项，则 Sybase Central 或 Interactive SQL 只会在该进程停止和重新启动时检测语言设置变化。

◆ 启用了快速启动程序选项时更改语言设置

1. 选择 [工具] » [选项]。
2. 在 [选项] 窗口的 [常规] 选项卡上，清除 [启用快速启动程序] 选项。
单击 [确定]。
3. 关闭 Sybase Central 或 Interactive SQL。
4. 根据需要更改语言设置。例如，运行以下命令将语言设置更改为德语：

```
dblang DE
```
5. 启动 Sybase Central 或 Interactive SQL。
6. 重新启用快速启动程序选项：
 - a. 选择 [工具] » [选项]。
 - b. 在 [选项] 窗口的 [常规] 选项卡，选择 [启用快速启动程序] 选项。
 - c. 单击 [确定]。

◆ 禁用了快速启动程序选项时更改语言设置

1. 关闭 Sybase Central 或 Interactive SQL。
2. 根据需要更改语言设置。例如，运行以下命令将语言设置更改为德语：

```
dblang de
```
3. 重新启动 Sybase Central 或 Interactive SQL。

或者，可通过关闭 `scjview` 或 `dbisql` 进程来停止快速启动程序。

另请参见

- [“SALANG 环境变量”一节第 355 页](#)
- [“使用快速启动程序选项”一节第 686 页](#)

示例

以下命令会显示包含当前设置的窗口：

```
dblang
```

以下命令将设置更改为德语，并显示包含以前设置和新设置的窗口：

```
dblang de
```

日志传送管理器实用程序 (dbltm)

读取数据库事务日志并将已提交的更改发送到复制服务器。

语法

dbltm [*options*]

选项	说明
@data	读取来自指定的环境变量或配置文件的选项。请参见“使用配置文件”一节第 695 页。 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“文件隐藏实用程序 (dbfhide)”一节第 724 页。
-A	防止过滤更新。缺省情况下，不复制维护用户所做的所有更改。如果设置了 -A 选项，则复制这些更改。此选项在非层次结构的复制服务器安装中（其中，数据库既充当复制站点又充当主站点）可能会很有用。
-C config-file	使用配置文件 <i>config-file</i> 以确定 LTM 设置。缺省配置文件是 <i>dbltm.cfg</i> 。请参见“LTM 配置文件”一节第 749 页。
-I interface-file	（大写的 I。）使用指定的接口文件。接口文件是由 DSEDIT 创建的文件，它包含 Open Server 的连接信息。缺省接口文件为 <i>SQL.ini</i> ，位于 Sybase 目录的 <i>ini</i> 子目录中。
-M	启动恢复操作。LTM 从最早的可用位置开始读取日志。如果在配置文件中指定脱机目录，则 LTM 从最早的脱机日志文件开始读取。
-S LTM-name	提供此 LTM 的服务器名称。缺省 LTM 名称为 DBLTM_LTM。LTM 名称必须对应于在 DSEDIT 中输入的 LTM 的 Open Server 名称。
-dl	显示 LTM 窗口中（或命令提示符处）以及日志文件（如果已指定）中的所有消息。
-ek key	直接在命令行上为高度加密的数据库指定加密密钥。如果您有一个高度加密的数据库，则必须提供加密密钥，才能使用数据库或事务日志，包括脱机事务日志。对于高度加密数据库，必须指定 -ek 或 -ep，但不要同时指定这两者。如果不为高度加密的数据库指定密钥，该命令将失败。
-ep	指定需要系统提示输入加密密钥。使用此选项将显示一个窗口，可以在其中输入加密密钥。加密密钥从不以明文显示，因此提供了额外的安全保证。对于高度加密数据库，必须指定 -ek 或 -ep，但不要同时指定这两者。如果不为高度加密的数据库指定密钥，该命令将失败。
-o filename	使用一个不同于缺省日志文件 (<i>dbltm.log</i>) 的日志文件。将日志传送操作的输出消息写入此文件。

选项	说明
-os size	指定输出文件的最大大小（以字节为单位）。最小值是 10000（一万）。如果日志文件增大到超过此限值，则将它重命名为 <i>yymddxx.ltm</i> 。对于在给定日期创建的每个文件，将递增 <i>yymddxx.ltm</i> 中 <i>xx</i> 的值。
-ot file	使用一个不同于缺省日志文件 (<i>dbltm.log</i>) 的日志文件，并在启动 LTM 时截断该日志文件（删除全部现有内容）。来自日志传送操作的输出消息将发送到此文件中，以供以后查看。
-q	在启动 LTM 时最小化窗口。
-s	记录由 LTM 生成的所有 LTL 命令。此选项只用来诊断问题，建议不要在生产环境中使用。使用此选项会大大降低性能。
-ud	在 Unix 操作系统上，可将 LTM 作为守护程序运行。如果以此方式运行，输出就会记录到日志文件中。
-ux	如果 <i>dbltm</i> 能在 Unix 操作系统上找到可用的显示，则打开 [日志传送管理器] 窗口。如果找不到（例如，因为没有设置 <code>DISPLAY</code> 环境变量或 X Windows Server 没有运行），则 <i>dbltm</i> 将无法启动。在 Microsoft Windows 上， <i>dbltm</i> 窗口将自动出现。
-v	为进行调试而显示除 LTL 消息之外的消息。

注释

日志传送管理器（Log Transfer Manager，简称 LTM）也称为 **replication agent**。任何作为主站点参与复制服务器安装的 SQL Anywhere 数据库都需要 LTM。

SQL Anywhere LTM 读取数据库事务日志并将已提交的更改发送到复制服务器。在复制站点上不需要 LTM。

LTM 使用名为日志传送语言（Log Transfer Language，简称 LTL）的语言将提交的更改发送到复制服务器。

缺省情况下，LTM 使用名为 *DBLTM.LOG* 的日志文件保存状态消息和其它消息。您可以使用选项更改此文件的名称，以及更改发送到该文件的消息的数量和类型。

退出代码是零（成功）或非零（失败）。请参见“[软件组件的退出代码](#)”一节《[SQL Anywhere 服务器 - 编程](#)》。

LTM 配置文件

SQL Anywhere 与 Adaptive Server Enterprise 的 LTM 配置文件非常类似。本节介绍了 SQL Anywhere LTM 配置文件中的条目，以及与 Adaptive Server Enterprise LTM 配置文件的不同之处。

LTM 使用的配置文件是使用 `-C` 选项指定的。

LTM 配置文件参数

下表介绍 LTM 可识别的每个配置参数。此列表中包括了由 Adaptive Server Enterprise LTM 使用而 SQL Anywhere LTM 未使用的选项，这些选项被标记为 [忽略]（在这种情况下，它们可能存在于配置文件中，但不起作用）或 [不支持]（在这种情况下，如果它们存在于配置文件中，则会导致错误）。

参数	说明
APC_pw	APC_user 登录名的口令。此条目只存在于 SQL Anywhere LTM 配置文件中。
APC_user	在主站点上执行异步过程时所使用的用户 ID。此用户 ID 必须对主站点上所有异步过程具有适当的权限。此条目只存在于 SQL Anywhere LTM 配置文件中。
backup_only	缺省情况下，设置为 off。如果设置为 on，则 LTM 只复制已备份的事务。
batch_ltl_cmds	设置为 on（缺省值），以使用批处理模式。批处理模式可以提高总体吞吐量，但可能会导致较长的响应时间。
batch_ltl_sz	当 batch_ltl_cmds 设置为 on 时，发送到复制服务器之前在缓冲区中保存的命令数。缺省值为 200。
batch_ltl_mem	当 batch_ltl_cmds 设置为 on 时，在将缓冲区的内容发送到复制服务器之前，缓冲区可以使用的内存量。缺省值为 256 KB。
Continuous	缺省情况下，设置为 on。当设置为 off 时，复制完所有已提交的数据后，LTM 将自动关闭。
LTM_admin_pw	LTM_admin_user 登录名的口令。
LTM_admin_user	用于登录到 LTM 的系统管理员 LTM 登录名。当有用户登录到 LTM 来关闭它时，需要有这个参数 LTM 才能检查该用户是否具有正确的登录名。
LTM_charset	供 LTM 使用的 Open Client/Open Server 字符集。
LTM_language	供 LTM 使用的 Open Client/Open Server 语言。
LTM_sortorder	LTM 用于比较用户名的 Open Client/Open Server 排序顺序。您可以指定与 LTM 的字符集兼容的任何受 Adaptive Server Enterprise 支持的排序顺序。复制系统中的所有排序顺序都应该相同。 缺省排序顺序是二进制排序。
maint_cmds_to_skip	忽略。

参数	说明
qualify_table_owners	将此参数设置为 on，以便 LTM 将带有表名、列名以及表所有者的 LTL 发送给复制服务器。此设置适用于所有复制表，而且创建复制定义语句必须与此设置匹配。缺省值是 off。
rep_func	设置为 on 以使用异步过程调用 (APC)。缺省值是 off。
Retry	连接 SQL Anywhere 数据库服务器或 Replication Server 失败后，重试连接之前等待的秒数。缺省值是 10 秒。
RS	LTM 向其传送日志的复制服务器的名称。
RS_pw	RS_user 登录名的口令。
RS_source_db	LTM 将其日志传送到复制服务器的数据库的名称。此名称必须与在复制服务器连接定义中定义的数据库名称匹配。大多数配置都使用相同的 RS_Source_db 和 SQL_database 配置选项。
RS_source_ds	LTM 将其日志传送到复制服务器的服务器名称。此名称必须与在复制服务器连接定义中定义的服务器名称匹配。大多数配置都对 RS_Source_ds 和 SQL_server 配置选项使用相同的设置。
RS_user	LTM 登录到 Replication Server 所使用的登录名。在复制服务器中，必须已经授予该登录名 connect source 权限。
scan_retry	LTM 在事务日志扫描之间等待的秒数。此参数的定义不同于 Adaptive Server Enterprise LTM。在记录到达日志中时，SQL Anywhere 服务器不会自动开始扫描日志。因此，您可能希望将 scan_retry 值设置为比 Adaptive Server Enterprise LTM 中此参数的值更小的数字。
skip_ltl_cmd_err	此参数会在发生 LTL 命令错误时通知复制代理是继续进行还是关闭。如果指定 skip_ltl_cmd_err=on，则复制代理显示导致发生错误的 LTL 命令，然后跳过 LTL 并继续进行复制。如果将此参数设置为 off，则复制代理显示导致发生错误的 LTL 命令，然后关闭。缺省情况下，此参数被设置为 off。
SQL_database	LTM 连接到的服务器 SQL_server 上的主站点数据库名。对于恢复期间的 Adaptive Server Enterprise，这是 LTM 将其日志传送到复制服务器的临时数据库。SQL Anywhere LTM 使用 SQL_log_files 参数来定位脱机事务日志。
SQL_log_files	保存脱机事务日志的目录。在 LTM 启动时该目录必须存在。此条目只存在于 SQL Anywhere LTM 配置文件中。
SQL_pw	SQL_user 用户 ID 的口令。

参数	说明
SQL_server	LTM 连接到的主站点 SQL Anywhere 服务器的名称。对于恢复期间的 Adaptive Server Enterprise，这是具有临时数据库（LTM 将其日志传送到复制服务器）的数据服务器。LTM 使用 SQL_log_files 参数来定位脱机事务日志。
SQL_user	LTM 使用该登录名来连接 RS_source_ds 和 RS_source_db 指定的数据库。

示例

下面是一个示例 LTM 配置文件。

```
# This is a comment line
# Names are case sensitive.
SQL_user=SA
SQL_pw=sysadmin
SQL_server=PRIMESV
SQL_database=primedb
RS_source_ds=PRIMEOS
RS_source_db=primedb
RS=MY_REP$ERVER
RS_user=sa
RS_pw=sysadmin
LTM_admin_user=DBA
LTM_admin_pw=sql
LTM_charset=cp850
scan_retry=2
SQL_log_files=e:\logs\backup
APC_user=sa
APC_pw=sysadmin
```

翻译日志文件实用程序 (dbtran)

将事务日志文件翻译为 SQL 命令文件。

语法

针对数据库服务器运行：

```
dbtran [ options ] -c { connection-string } -n SQL-file
```

针对事务日志运行：

```
dbtran [ options ] [ transaction-log ] [ SQL-file ]
```

选项	说明
@data	<p>读取来自指定的环境变量或配置文件的选项。请参见“使用配置文件”一节第 695 页。</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“文件隐藏实用程序 (dbfhide)”一节第 724 页。</p>
-a	<p>控制是否在事务日志中显示未提交的事务。</p> <p>事务日志包含任何事务在最近一次执行 COMMIT 之前做出的更改。最近一次提交之后做出的更改不会出现在事务日志中。</p> <p>如果没有指定 -a，则输出文件中只显示已提交的事务。如果指定 -a，则会输出事务日志中存在的任何未提交事务，并且后跟一条 ROLLBACK 语句。</p>
-c "keyword=value; ..."	<p>在针对数据库服务器运行该实用程序时，此选项用于指定连接字符串。请参见“连接参数”一节第 248 页。</p>
-d	<p>指定按照从最早到最晚的次序写入事务。提供此功能的主要目的是在审计数据库活动时使用：不能针对数据库应用 dbtran 的输出。</p>
-ek key	<p>为高度加密的数据库指定加密密钥。如果您有一个高度加密的数据库，则必须提供加密密钥才能使用数据库或事务日志。</p> <p>对于高度加密数据库，您必须指定 -ek 或 -ep，但不要同时指定这两者。如果没有指定正确的加密密钥，则该命令将失败。</p> <p>如果要针对数据库服务器运行 dbtran（使用 -c 选项），请使用连接参数而非 -ek 选项来指定密钥。例如，以下命令用于从数据库服务器 sample 中获取有关数据库 <i>enc.db</i> 的事务日志信息，并将其输出保存到 <i>log.sql</i> 中。</p> <pre>dbtran -n log.sql -c "ENG=sample;DBF=enc.db;UID=DBA;PWD=sql;DBKEY=mykey"</pre>

选项	说明
-ep	<p>提示输入加密密钥。使用此选项将显示一个窗口，可以在其中输入加密密钥。加密密钥从不以明文显示，因此提供了额外的安全保证。</p> <p>对于高度加密数据库，您必须指定 -ek 或 -ep，但不要同时指定这两者。如果没有指定正确的加密密钥，则该命令将失败。</p> <p>如果要针对数据库服务器运行 dbtran（使用 -c 选项），请使用连接参数而非 -ep 选项来指定密钥。例如，以下命令用于从数据库服务器 sample 中获取有关数据库 enc.db 的事务日志信息，并将其输出保存到 log.sql 中。</p> <pre>dbtran -n log.sql -c "ENG=sample;DBF=enc.db; UID=DBA;PWD=sql;DBKEY=mykey"</pre>
-f	只输出上一个检查点之后完成的事务。
-g	<p>如果启用了审计数据库选项，则审计信息会添加到事务日志中。使用此选项，可将审计信息作为注释包括在输出文件中。请参见“auditing 选项 [数据库]”一节第 484 页。</p> <p>-g 选项隐含 -a、-d 和 -t 选项。</p>
-ir offset1,offset2	输出在两个指定偏移之间的事务日志部分。
-is source,...	<p>输出对行（以下一个或多个源中的操作已修改了这些行）的操作，这些源被指定为逗号分隔的列表：</p> <ul style="list-style-type: none"> ● All 所有行。这是缺省设置。 ● SQLRemote 只包括使用 SQL Remote 修改的行。您也可以使用简写形式 SR。 ● RepServer 只包括使用复制代理 (LTM) 和复制服务器修改的行。您也可以使用简写形式 RS。 ● Local 只包括没有复制的行。
-it owner.table,...	输出对位于逗号分隔的列表中的并且是指定的表所执行的操作。应该以 <i>owner.table</i> 的形式指定每个表。
-j date/time	只翻译给定日期和/或时间之前的最后一次检查点操作以来的事务。用户提供的参数可以用引号括起来的日期、时间或时间及日期。如果省略时间，则将时间假定为一天的起始时间。如果省略日期，则假定为当天。以下是可接受的日期及时间格式：“YYYY/MMM/DD HH:NN”。
-k	防止检测到错误时消除部分 <i>.sql</i> 文件。如果在 dbtran 运行期间检测到错误，通常会消除到该时刻为止所生成的 <i>.sql</i> 文件，以确保不会因不小心而使用部分文件。如果试图从损坏的事务日志中保留事务，指定此选项将甚为有用。

选项	说明
-m	指定包含事务日志的目录。此选项必须与 -n 选项联合使用。
-n filename	在针对数据库服务器运行 dbtran 实用程序时，使用此选项指定保存 SQL 语句的输出文件。
-o filename	将输出消息写入指定的文件中。
-q	以安静模式运行—不显示消息。
-r	删除所有未提交的事务。这是缺省行为。
-rsu username,...	指定以逗号分隔的用户名列表以替换缺省的复制服务器用户名。缺省情况下， -is 选项假定缺省的复制服务器用户名为 dbmaint 和 sa 。
-s	控制如何生成 UPDATE 语句。如果不使用此选项，而且在表上没有主键或唯一索引，则在有重复行的情况下，翻译日志文件实用程序生成带有非标准 FIRST 关键字的 UPDATE 语句。如果使用此选项，则忽略 FIRST 关键字，以便与 SQL 标准兼容。
-sr	将生成的用以说明 SQL Remote 如何将操作分发到远程站点的注释保存到输出文件中。
-t	控制是否在命令文件中包含触发器。缺省情况下，触发器执行的操作不包括在命令文件中。如果在数据库中有匹配的触发器，则在针对该数据库运行命令文件时，触发器将自动执行操作。如果运行命令文件的数据库中不存在匹配触发器，则应该包括触发器操作。
-u userid,...	将事务日志的输出限制为只包括指定的用户。
-x userid,...	将事务日志的输出限制为不包括指定的用户。
-y	在不提示进行确认的情况下替换现有的命令文件。如果您指定了 -q ，则还必须指定 -y ，否则操作将失败。
-z	将触发器生成的事务只作为注释包括在输出文件中。
<i>transaction-log</i>	指定要翻译的日志文件。不能与 -c 或 -m 选项一起使用。
<i>SQL-file</i>	命名包含已翻译信息的输出文件。只与 <i>transaction-log</i> 一起使用。

注释

dbtran 实用程序获取事务日志中的信息，并将该信息作为一组 SQL 语句和注释置于输出文件中。该实用程序可以按照以下方式运行：

- **针对数据库服务器** 如果 dbtran 针对数据库服务器运行，则该实用程序为标准客户端应用程序。它使用在 -c 选项之后指定的连接字符串连接到数据库服务器，并将输出保存在使用 -n 选项指定的文件中。以此方式运行需要有 DBA 权限。

以下命令用于翻译来自服务器 **demo11** 的日志信息，并将输出保存在名为 *demo.sql* 的文件中。

```
dbtran -c "ENG=demo11;DBN=demo;UID=DBA;PWD=sql" -n demo.sql
```

- **针对事务日志文件** dbtran 针对事务日志运行时，该实用程序直接作用于事务日志文件。如果不想让用户拥有运行此语句的能力，则应保护事务日志文件以禁止一般性访问。

```
dbtran demo.log demo.sql
```

dbtran 实用程序运行时，它会显示事务日志中最早的日志偏移。这可以作为确定多个日志文件的生成顺序的有效方法。

如果使用 -c，dbtran 将尝试翻译联机事务日志文件及其同一目录中的所有脱机事务日志文件。如果目录中包含不止一个数据库的事务日志文件，dbtran 可能会给出错误消息。为了避免此问题，请确保每个目录中包含的是仅针对一个数据库的事务日志文件。

一个事务可以跨多个事务日志。如果事务日志文件包含跨多个日志的事务，则只翻译一个事务日志文件（例如，dbtran demo.log）可能会导致跨日志的事务丢失。为了让 dbtran 生成完整的事务，请将 -c 或 -m 选项与目录中的事务日志文件一起使用。请参见“[通过多个事务日志恢复数据库](#)”一节第 837 页。

您可以以下列方式访问翻译日志文件实用程序。

- 从 Sybase Central 使用 [\[翻译日志文件向导\]](#)。
- 在命令提示符处，使用 dbtran 命令。如果要合并到批处理文件或命令文件中，此命令比较有用。

退出代码是零（成功）或非零（失败）。请参见“[软件组件的退出代码](#)”一节《[SQL Anywhere 服务器 - 编程](#)》。

Ping 实用程序 (dbping)

查找数据库服务器，并测试数据库连接。

语法

dbping [*options*]

选项	说明
@data	<p>读取来自指定的环境变量或配置文件的选项。请参见“使用配置文件”一节第 695 页。</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“文件隐藏实用程序 (dbfhide)”一节第 724 页。</p>
-c "keyword=value; ..."	<p>指定控制 dbping 行为的连接参数。如果未指定连接参数，则使用来自 SQLCONNECT 环境变量的连接参数（如果已设置）。请参见“连接参数”一节第 248 页。</p> <p>如果使用以下命令启动 dbping，且有一个名为 demo11 的数据库服务器已经在运行，则 dbping 会尝试连接到名为 demo 的数据库。如果没有这样的数据库该数据库服务器上运行，则数据库服务器会尝试装载 demo.db。如果找不到名为 demo11 的服务器，dbping 会尝试自动启动一个服务器。</p> <pre>dbping -d -c "UID=DBA;PWD=sql;ENG=demo11;DBN=demo;DBF=samples-dir \demo.db"</pre> <p>有关 samples-dir 的信息，请参见“示例目录”一节第 368 页。</p>
-d	<p>强制数据库回应，而不仅仅是强制服务器回应。</p> <p>如果提供 -d 选项，则 dbping 只有在既连接到服务器又连接到数据库的情况下才会报告成功。如果不提供 -d 选项，则 dbping 会在找到 -c 选项所指定的服务器时就报告成功。</p> <p>例如，如果您拥有的名为 blair 的数据库服务器正在运行示例数据库，则以下命令将成功：</p> <pre>dbping -c "ENG=blair;DBN=demo"</pre> <p>以下命令将失败，并显示消息 [强制数据库回应失败 -- 未找到指定的数据库]：</p> <pre>dbping -c "ENG=blair;DBN=demo" -d</pre>
-en	<p>指定您希望在指定的任一属性返回 NULL 时，dbping 退出并返回失败的返回代码。缺省情况下，如果由 -pc、-pd 或 -ps 指定的属性的值为未知，则 dbping 会输出 NULL 并退出，同时返回成功的返回代码。此选项只能与 -pc、-pd 和 -ps 一起使用。</p>

选项	说明
-l library	<p>指定要使用的库（不带其文件扩展名）。此选项避免了使用 ODBC 驱动程序管理器，因此在 Unix 操作系统上特别有用。</p> <p>例如，以下命令行直接装载 ODBC 驱动程序：</p> <pre>dbping -m -c "DSN=SQL Anywhere 11 Demo" -l dbodbc11</pre> <p>在 Unix 上，如果您希望使用线程连接库，则必须使用 Ping 实用程序的线程版本：<code>dbping_r</code>。</p>
-m	<p>使用 ODBC 建立一个连接。缺省情况下，实用程序使用嵌入式 SQL 接口进行连接。</p>
-o filename	<p>将输出消息写入指定的文件中。</p>
-pc property,...	<p>显示指定的连接属性。在以逗号分隔的列表中提供这些属性。如果使用此选项，则您必须指定足够的连接信息以建立数据库连接。请参见“连接属性”一节第 566 页。</p> <p>例如，以下命令将显示作为连接属性提供的 <code>fire_triggers</code> 选项设置。</p> <pre>dbping -c ... -pc fire_triggers</pre>
-pd property[@db-name],...	<p>显示指定的数据库属性。在以逗号分隔的列表中提供这些属性。请参见“数据库属性”一节第 601 页。</p> <p>例如，以下命令显示数据库正在使用的页面大小：</p> <pre>dbping -c ... -pd PageSize</pre> <p>另外，也可以指定正在数据库服务器上运行并想要从中获取值的数据库的名称。对于列出的每个属性，如果未通过将 <code>@db-name</code> 附加到属性来指定数据库名称，则将使用前一个属性所使用的数据库名称。</p> <p>以下命令显示数据库 <code>mydb</code> 的页面大小和归类：</p> <pre>dbping -c ... -pd PageSize@mydb,Collation</pre>
-ps property,...	<p>显示指定的数据库服务器属性。在以逗号分隔的列表中提供这些属性。如果使用此选项，则您必须指定足够的连接信息以建立数据库连接。请参见“数据库服务器属性”一节第 588 页。</p> <p>例如，以下命令显示数据库服务器的许可存取设备数或处理器数：</p> <pre>dbping -c ... -ps LicenseCount</pre>
-q	<p>以安静模式运行—不显示消息。</p>

选项	说明
-s	返回有关运行 dbping 的计算机与运行数据库服务器的计算机之间的网络性能的信息。将显示大致的连接速度、等待时间和吞吐量。要指定连接到某服务器上的数据库的连接参数，通常需要 -c 选项。对于嵌入式 SQL 连接，只能使用 dbping -s。如果另外指定了 -m 或 -l，则将忽略此选项。缺省情况下，对于需要测定的每项统计信息，dbping -s 都会循环执行请求，时间至少为一秒钟。但是，最多只能重复执行 200 次的连接和断开操作而不管花费的时间有多少，这样能够避免占用过多的资源。在速度较慢的网络上，即使是达到针对每项统计信息的最少重复执行次数可能也得需要几秒钟的时间。性能统计信息是近似的结果，当客户端和服务器计算机均完全闲置时，得到的统计结果会更精确。请参见“ 测试嵌入式 SQL 连接的性能 ”一节第 130 页。
-st time	此选项与 -s 相同，但它能指定时间长度（以秒为单位），对于需要测定的每项统计信息，dbping 会在该段时间内循环执行请求。与 -s 相比，此选项能够获得更精确的计时信息。请参见“ 测试嵌入式 SQL 连接的性能 ”一节第 130 页。
-z	显示用于尝试连接的网络通信协议和其它诊断消息。只有在尝试进行嵌入式 SQL 连接时，才能使用此选项。即无法与 -m 或 -l 组合使用。

注释

dbping 实用程序是帮助解决连接问题的工具。它接受完整或部分连接字符串，然后返回消息，以指出定位服务器或数据库或者建立连接的尝试是否成功。

该实用程序可用于嵌入式 SQL 连接或 ODBC 连接。它不能用于 jConnect (TDS) 连接。

退出代码是零（成功）或非零（失败）。请参见“[软件组件的退出代码](#)”一节《[SQL Anywhere 服务器 - 编程](#)》。

重建实用程序 (rebuild)

重建数据库文件。

语法

```
rebuild old-database new-database [ DBA-password ]
```

注释

此批处理文件或 shell 脚本使用 `dbunload` 将 *old-database* 重建为 *new-database*。指定两个数据库名时都不应带扩展名。扩展名 `.db` 会自动被添加。

如果 *old-database* 中 DBA 用户的口令不是初始口令 `sql`，则必须指定 *DBA-password*。

Rebuild 使用 `-an` 选项运行 `dbunload` 命令。

您还可以通过 Sybase Central 中的 [\[卸载数据库向导\]](#)，作为卸载过程的一部分来重建数据库。请参见“使用 [\[卸载数据库向导\]](#) 导出数据”一节《[SQL Anywhere 服务器 - SQL 的用法](#)》。

退出代码是零（成功）或非零（失败）。请参见“[软件组件的退出代码](#)”一节《[SQL Anywhere 服务器 - 编程](#)》。

此实用程序不支持使用 `@data` 参数从配置文件中读入选项。

另请参见

- [“卸载实用程序 \(dbunload\)”](#) 一节第 795 页
- [“初始化实用程序 \(dbinit\)”](#) 一节第 729 页
- [“Interactive SQL 实用程序 \(dbisql\)”](#) 一节第 740 页

脚本执行实用程序 (dbrunsql)

允许您针对在 Windows Mobile 上运行的数据库，执行 SQL 命令，并运行命令文件。

语法

dbrunsql [*options*] [*SQL-script-file* | *SQL-command*]

选项	说明
-c " <i>keyword=value; ...</i> "	指定连接参数。请参见“ 连接参数 ”一节第 248 页。
-d	将从结果集中导出的数据写入输出文件中。如果不指定 -d ，则所有 dbrunsql 输出均会写入输出文件中。
-e [c p s]	控制在语句执行过程中遇到错误时的行为。缺省情况下， dbrunsql 会在出现错误时提示用户。 <ul style="list-style-type: none"> ● c 忽略错误并继续执行语句。 ● p 给用户以提示，以了解用户是否要继续。 ● s 停止执行语句。
-f [f a]	指定 dbrunsql 在导出结果集时使用的数据格式。可以指定以下各值之一： <ul style="list-style-type: none"> ● a 导出数据时使用 ASCII 格式。 ● f 导出数据时使用 FIXED 格式。这是缺省格式。
-g [+ -]	防止显示 GUI。缺省情况下会显示 dbrunsql GUI。
-o <i>filename</i>	将输出消息写入指定的文件中。
-q	取消输出消息。只有在使用命令或命令文件启动 dbrunsql 时，此选项才有意义。指定此选项不会取消错误消息，但会取消以下各项： <ul style="list-style-type: none"> ● 警告和其它非致命错误消息 ● 结果集的输出
-qc	在命令或脚本文件执行后立即关闭 dbrunsql 窗口。
-s <i>数字</i>	指定在使用 FIXED 格式导出结果集时，每列读取的最大字节数。缺省值是 255。
-v	在 dbrunsql 输出中包含每个 SQL 语句的所有行。否则，在执行脚本文件时，只显示当前所执行行的号码。

注释

dbrunsql 实用程序允许您针对某个数据库执行 SQL 命令或运行命令文件。只有 Windows Mobile 支持 SQL Anywhere 脚本执行实用程序 (dbrunsql)。

服务器枚举实用程序 (dblocate)

定位 TCP/IP 网络上的数据库服务器。

语法

dblocate [*options*] [*server-name*]

选项	说明
@data	<p>读取来自指定的环境变量或配置文件的选项。请参见“使用配置文件”一节第 695 页。</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“文件隐藏实用程序 (dbfhide)”一节第 724 页。</p>
-d	<p>对于找到的每个服务器，列出服务器名称和地址，随后是运行于该服务器上的数据库的以逗号分隔的列表。如果该列表超过 160 个字符，它将被截断并以省略号 (...) 结束。</p> <p>在 SQL Anywhere 9.0.2 及早期数据库服务器上运行的数据库，或以 -dh 数据库选项启动的数据库不会列出。请参见“-dh 数据库选项”一节第 237 页。</p>
-dn database-name	<p>对于正在运行指定名称的数据库的服务器，列出其服务器名称和地址。如果该列表超过 160 个字符，它将被截断并以省略号 (...) 结束。</p> <p>在 SQL Anywhere 9.0.2 及早期数据库服务器上运行的数据库，或以 -dh 数据库选项启动的数据库不会列出。请参见“-dh 数据库选项”一节第 237 页。</p>
-dv	<p>对于找到的每个服务器，显示服务器名称和地址，并在单独的行上列出在该服务器上运行的每个数据库。该列表不会被截断，所以此选项可用于显示那些使用 -d 选项时被截断的列表。</p> <p>在 SQL Anywhere 9.0.2 及早期数据库服务器上运行的数据库，或以 -dh 数据库选项启动的数据库不会列出。请参见“-dh 数据库选项”一节第 237 页。</p>
-n	<p>在输出中列出 IP 地址而非计算机名。这可提高性能，因为相对来说，查找计算机名可能会比较慢。</p>
-o filename	<p>将输出消息写入指定的文件中。</p>

选项	说明
-p <i>port-number</i>	仅显示使用指定 TCP/IP 端口号的服务器的名称和地址。TCP/IP 端口号必须介于 1 和 65535 之间。
-q	以安静模式运行—不显示消息。
-s <i>name</i>	仅显示具有指定服务器名称的服务器的名称和地址。如果使用此选项，就不能使用 -ss 选项（如果同时使用这两者，则可能找不到匹配的服务器）。
-ss <i>substr</i>	仅显示服务器名称中包含指定子串的服务器的名称和地址。如果使用此选项，就不能使用 -s 选项（如果同时使用这两者，则可能找不到匹配的服务器）。
-v	显示完整的服务器名称。缺省情况下，dblocate 将截断长度超过 40 个字节的数据库服务器名称。 9.0.2 版及早期版本的客户端（包括 dblocate）不能连接到名称长于 40 个字节的 10.0.0 版及更高版本的数据库服务器。
<i>server-name</i>	仅列出在具有指定 IP 地址或主机名的计算机上运行的数据库服务器。例如，以下命令将查找计算机 jfrancis 上的服务器： <code>dblocate jfrancis</code> 主机名或 IP 地址可以为任意格式，不管是否指定 -n 。例如，假定某个服务器正在 myhost.mycompany.com 上运行，其 IP 地址为 1.2.3.4。要从使用 mycompany.com 域的任何计算机中仅列出在此计算机上运行的服务器，可以使用 dblocate myhost、dblocate myhost.mycompany.com 或 dblocate 1.2.3.4 中的任何一个。

注释

服务器枚举实用程序 (dblocate) 用于定位通过 TCP/IP 在直接网络上运行的任何 SQL Anywhere 数据库服务器，并输出一份包含数据库服务器及其地址的列表。此列表包括替代服务器名。请参见“[-sn 数据库选项](#)”一节第 242 页。

根据网络的不同，dblocate 可能要在几秒钟之后才能打印出其结果。

注意

在 Mac OS X 上，如果数据库服务器正在使用 2638 之外的 TCP/IP 端口，则即使已使用 **-p** 选项指定该 TCP/IP 端口，dblocate 也无法找到它。请参见“[ServerPort 协议选项 \[PORT\]](#)”一节第 303 页。

退出代码是零（成功）或非零（失败）。请参见“[软件组件的退出代码](#)”一节《[SQL Anywhere 服务器 - 编程](#)》。

数据库服务器可将自身注册到跟踪企业中所有服务器的 LDAP 服务器上。这样客户端和 dblocate 都可以通过防火墙找到它们，无论它们是在 WAN 上还是 LAN 上，且无需指定 IP 地址。LDAP 仅与 TCP/IP 一起使用，而且仅在网络服务器上使用。请参见“[使用 LDAP 服务器进行连接](#)”一节第 138 页。

如果多次发现同一个数据库服务器名称，则 dblocate 会显示每个主机的 IP 地址，即使没有指定 -n 选项。在以下情况下可能会出现服务器重名现象：服务器运行在具有多个 IP 地址的计算机上（例如，计算机有多块网卡），或网络服务器运行在远程计算机上而同名的个人服务器运行在本地计算机上。

服务器授权实用程序 (dblic)

将软件许可应用于 SQL Anywhere 数据库服务器或 MobiLink 服务器。

语法

```
dblic [ options ] license-file "user-name" "company-name"
```

选项	说明
@data	读取来自指定的环境变量或配置文件的选项。请参见“使用配置文件”一节第 695 页。 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“文件隐藏实用程序 (dbfhide)”一节第 724 页。
-l type	指定在与在软件许可协议中所述的授权模式相匹配的许可类型。支持以下许可类型： <ul style="list-style-type: none"> ● 每存取设备 每存取设备许可会限制与数据库服务器连接的客户端数量。使用每存取设备授权时，网络数据库服务器将使用计算机上所有可用的 CPU，除非网络数据库服务器受 -gt 选项或正在运行的版本的限制。个人服务器仅限使用一个 CPU。 ● 处理器 处理器许可会限制那些可由数据库服务器使用的单独物理处理器的数量。数据库服务器可以使用的 CPU 数量可能进一步受到 -gt 选项或正在运行的 SQL Anywhere 版本的限制。个人数据库服务器仅限使用一个 CPU。 <p>对于此许可类型而言，数据库服务器将每个物理处理器都视为一个 CPU，而不将双核或超线程处理器视为多个处理器。如果拥有处理器许可，则对连接到数据库服务器的客户端的数量没有任何限制。</p>
-o filename	将输出消息写入指定的文件中。
-q	以安静模式运行—不显示消息。
-u license-number	指定许可的用户或处理器的总数。如果您是在增加额外的许可，那么此值代表总许可数，而非增加的许可数。
license-file	指定您要授权的个人数据库服务器、网络数据库服务器或 MobiLink 服务器的可执行文件或者许可文件的路径和文件名。 通过仅输入许可文件名，即可查看服务器可执行文件的当前许可信息。
user-name	指定许可的用户名。在数据库服务器启动时，此名称出现在数据库服务器消息窗口上。如果名称中有空格，则需要用双引号将名称括起来。
company-name	指定许可的公司名称。在数据库服务器启动时，此名称出现在数据库服务器消息窗口上。如果名称中有空格，则用双引号将名称括起来。

注释

服务器授权实用程序用于向 SQL Anywhere 数据库服务器或 MobiLink 服务器添加获得授权的用户或处理器。您必须仅按照您的许可协议来使用此实用程序，授权您有权授予的用户数或处理器数。运行此命令并不授予您许可。数据库服务器可以使用的 CPU 数量还可能会受到 SQL Anywhere 版本或 -gt 服务器选项的影响。请参见：

- “版本和授权”一节 《SQL Anywhere 11 - 简介》
- “-gt 服务器选项”一节第 185 页

此实用程序还将修改启动时由个人或网络数据库服务器和 MobiLink 服务器显示的用户名和公司名。通过仅输入许可文件的名称，还可使用此实用程序来查看个人或网络数据库服务器的当前许可信息。

许可信息保存在 .lic 文件中，该文件与服务器可执行文件位于相同的目录中。服务器会查找与正在运行的可执行文件的基本文件名相同的 .lic 文件。例如，如果数据库服务器可执行文件的名称是 *myserver.exe*，那么服务器就会查找名为 *myserver.lic* 的许可文件。缺省值情况下，会使用以下名称：

可执行文件	许可文件的名称
SQL Anywhere 个人数据库服务器 (dbeng11)	<i>dbeng11.lic</i>
SQL Anywhere 网络数据库服务器 (dbsrv11)	<i>dbsrv11.lic</i>
MobiLink 服务器 (mlsrv11)	<i>mlsrv11.lic</i>

当试图启动服务器时，如果相应的 *lic* 文件不可用，服务器将不会启动。许可文件是由 SQL Anywhere 的安装程序创建的。dblic 实用程序仅修改现有许可；它不创建新的许可文件。

退出代码是零（成功）或非零（失败）。请参见“软件组件的退出代码”一节 《SQL Anywhere 服务器 - 编程》。

在 Unix 上，数据库服务器可执行文件在缺省情况下不可写，因此使用服务器授权 (dblic) 实用程序会失败。在使用服务器授权实用程序之前，请确保可执行文件是可写的（例如，使用 `chmod +w`）。

有关 SQL Anywhere 授权的详细信息，请访问 <http://www.sybase.com/detail?id=1056242>。

示例

以下命令在数据库服务器可执行文件所在的同一目录中执行，可将 50 个用户的许可（其用户名为 Sys Admin，公司名为 My Co）应用于 Microsoft Windows 网络数据库服务器。必须在一行上输入该命令的所有内容：

```
dblic -l perseat -u 50 dbsrv11.lic "Sys Admin" "My Co"
```

以下消息将出现在屏幕上，表明许可成功：

```
授权的节点: 50 用户: Sys Admin 公司: My Co
```

以下命令将返回关于数据库服务器许可的信息：

```
dblic dbsrv11.lic
```

用于 Linux 的服务实用程序 (dbsvc)

创建、修改和删除 SQL Anywhere 服务。

语法

dbsvc [*modifier-options*] **-d** *svc*

dbsvc [*modifier-options*] **-g** *svc*

dbsvc [*modifier-options*] **-l**

dbsvc [*modifier-options*] **-status** *svc*

dbsvc [*modifier-options*] **-u** *svc*

dbsvc [*modifier-options*] *creation-options* **-w** *svc details*

dbsvc [*modifier-options*] **-x** *svc*

details:

full-executable-path [*options*]

主选项	说明
-d <i>service-name</i>	从服务列表中删除指定的服务。如果提供 -y ，则服务会在无确认的情况下被删除。
-g <i>service-name</i>	列出服务的定义。
-l	列出可用的 SQL Anywhere 服务。
-u <i>service-name</i>	启动名为 <i>service-name</i> 的服务。
-w <i>executable parameters</i>	<p>创建新服务，或者在存在同名服务的情况下覆盖该服务。如果提供 -y，则现有服务会在无确认的情况下被覆盖。</p> <p>您必须提供适用于正在创建的服务的命令行参数。</p> <p>请参见：</p> <ul style="list-style-type: none"> ● dbsrv11 和 dbeng11 “SQL Anywhere 数据库服务器” 一节第 148 页 ● mlsrv11 “MobiLink 服务器选项” 《MobiLink - 服务器管理》 ● dbmlsync “dbmlsync 语法” 一节 《MobiLink - 客户端管理》 ● dbremote “消息代理 (dbremote)” 一节 《SQL Remote》
-x <i>service-name</i>	停止名为 <i>service-name</i> 的服务。

创建选项	说明
-a acct	所有服务都用一个 Linux 帐户运行。如果用您事先创建的帐户运行，则必须使用 -a 选项来指定该帐户。 除守护程序帐户之外，其它所有帐户均需要 "作为服务登录" 的特权。
-as	所有服务都用一个 Linux 帐户运行。如果指定 -as ，服务将用 Linux 守护程序帐户运行。不需要口令。 -a 和 -as 必须要使用一个。
-od	指定系统信息文件的位置（如果需要）。
-pr	为 Linux 过程设置运行级别。
-rl	指定启动服务的运行级别。
-rs	在创建服务时指定服务依赖性。
-s	为 SQL Anywhere 服务设置启动行为。可以将启动行为设置为 [自动] 或 [手动]。缺省值为 [手动]。
-status	返回服务运行的状态。
-t type	指定此服务的类型。您可以从以下类型中进行选择： <ul style="list-style-type: none"> ● Network SQL Anywhere 网络数据库服务器 (dbsrv11)。请参见“SQL Anywhere 数据库服务器”一节第 148 页。 ● Standalone SQL Anywhere 个人数据库服务器 (dbeng11)。请参见“SQL Anywhere 数据库服务器”一节第 148 页。 ● DBRemote SQL Remote 消息代理 (dbremote)。请参见“消息代理 (dbremote)”一节《SQL Remote》。 ● MobiLink MobiLink 服务器 (mlsrv11)。请参见“mlsrv11 语法”一节《MobiLink - 服务器管理》。 ● dbmlsync MobiLink 同步客户端 (dbmlsync)。请参见“dbmlsync 语法”一节《MobiLink - 客户端管理》。 <p>所有服务类型的缺省设置为 [Standalone]。</p>

修饰符选项	说明
-cm	显示用于创建服务的命令。此选项可用于将创建命令输出到一个文件中，随后可利用它在其它计算机上添加该项服务，或是在对服务做了更改时，用它来将服务恢复到其原始状态。在使用 -cm 时，必须指定 -g 选项或 -l 选项，否则命令将失败。指定 -g 会显示指定服务的创建命令，而指定 -l 会显示所有服务的创建命令。
-q	取消向控制台的消息输出。如果您在修改或删除一个现有服务时指定此选项，则还必须指定 -y ，否则操作将失败。
-y	自动执行操作，不提示您进行确认。此选项可以与 -w 或 -d 选项一起使用。如果您在修改或删除一个现有服务时指定 -q ，则还必须指定 -y ，否则操作将失败。

注释

服务使用一组选项运行数据库服务器或其它应用程序。此实用程序为 Linux 上 SQL Anywhere 服务的管理提供了一套完整的方法。

因为服务通常在不同的环境中运行，建议您在创建服务时用完整的路径指定数据库文件名。另外还建议您不要在数据源名称中使用空格。

与大多数 Linux 服务一样，dbsvc 实用程序在 `/etc/init.d` 中创建服务文件。服务的命名约定是 `SA_service-name`。例如，如果创建了名为 `myserv` 的服务，则可以发出以下命令来启动该服务：

```
/etc/init.d/SA_myserv start
```

以下命令可获取该服务的状态：

```
/etc/init.d/SA_myserv status
```

以下命令可返回该服务的使用信息：

```
/etc/init.d/SA_myserv
```

示例

创建名为 `myserv` 的个人服务器服务，它使用指定的参数启动指定的服务器。该服务器以 `LocalSystem` 用户身份运行：

```
dbsvc -as -w myserv -n myeng -c 8m "/tmp/demo.db"
```

创建名为 `mynetworkserv` 的网络服务器服务。该服务器使用本地帐户运行，并在重新启动计算机时自动启动：

```
dbsvc -as -t network -w mynetworkserv -x tcpip -c 8m "/tmp/demo.db"
```

列出关于 `myserv` 服务的所有详细信息：

```
dbsvc -g myserv
```

删除名为 `myserv` 的服务，但不提示您进行确认：

```
dbsvc -y -d myserv
```

创建名为 `mysyncservice` 的服务:

```
dbsvc -as -t dbmlsync -o syncinfo.txt -w mysyncservice -c "/tmp/CustDB.db"
```

生成用以创建 `service_1` 服务的命令并将其输出到控制台:

```
dbsvc -cm -g service_1
```

控制台包含以下内容:

```
'dbsvc -t Standalone -as -y -w "service_1" -n'
```

使用 `dbsvc` 来启动服务:

```
dbsvc -u myserv
```

使用 `dbsvc` 来停止服务:

```
dbsvc -x myserv
```

使用 `dbsvc` 来获取服务的状态:

```
dbsvc -status myserv
```

用于 Windows 的服务实用程序 (dbsvc)

创建、修改和删除 SQL Anywhere 服务。

语法

dbsvc [*modifier-options*] **-d** *svc*

dbsvc [*modifier-options*] **-g** *svc*

dbsvc [*modifier-options*] **-l**

dbsvc [*modifier-options*] **-u** *svc*

dbsvc [*modifier-options*] *creation-options* **-w** *svc details*

dbsvc [*modifier-options*] **-x** *svc*

details:

<*full-executable-path*> [*options*]

主选项	说明
@data	读取来自指定的环境变量或配置文件的选项。请参见“ 使用配置文件 ”一节第 695 页。 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“ 文件隐藏实用程序 (dbfhide) ”一节第 724 页。
-d service-name	从服务列表中删除指定的服务。如果提供 -y ，则服务会在无确认的情况下被删除。
-g service-name	列出服务的定义，但不包括口令。
-l	列出可用的 SQL Anywhere 服务。
-u service-name	启动名为 <i>service-name</i> 的服务。

主选项	说明
-w <i>executable parameters</i>	<p>创建新服务，或者在存在同名服务的情况下覆盖该服务。如果提供 -y，则现有服务会在无确认的情况下被覆盖。</p> <p>您必须提供希望用作服务的可执行文件的完整路径，因为运行服务所使用的帐户在其路径中可能没有相应的 SQL Anywhere 安装目录。</p> <p>您必须为正在创建的服务提供适当的参数。</p> <p>请参见：</p> <ul style="list-style-type: none"> ● dbsrv11 和 dbeng11 “SQL Anywhere 数据库服务器” 一节第 148 页 ● mlsrv11 “MobiLink 服务器选项” 《MobiLink - 服务器管理》 ● dbmlsync “dbmlsync 语法” 一节 《MobiLink - 客户端管理》 ● dblsn “用于 Windows 设备的监听器实用程序” 一节 《MobiLink - 服务器启动的同步》 ● dbremote “消息代理 (dbremote)” 一节 《SQL Remote》 ● dbns “广播转发器实用程序 (dbns11)” 一节第 703 页 ● dbltm “日志传送管理器实用程序 (dbltm)” 一节第 748 页 ● rshost “中继服务器状态管理器” 一节 《MobiLink - 服务器管理》 ● RSOE “出站启动器” 一节 《MobiLink - 服务器管理》 ● SAVSSWriter “使用 SQL Anywhere 卷影复制服务 (VSS)” 一节第 829 页
-x <i>service-name</i>	停止名为 <i>service-name</i> 的服务。

创建选项	说明
-a <i>acct</i>	<p>命名 Microsoft Windows 帐户。所有服务都用一个 Microsoft Windows 帐户运行。如果以已经创建的帐户运行，则必须通过 -a 选项指出该帐户名，并通过 -p 选项提供口令。</p> <p>除 LocalSystem 帐户外的所有其它帐户都需要有 "作为服务登录" 的特权。如果帐户未启用 "作为服务登录" 的特权，系统会提示您启用该特权。如果同时还指定 -y 选项，dbsvc 会尝试在不予提示的情况下授予您 "作为服务登录" 的特权。如果指定 -q 选项但不指定 -y 选项，您不会收到授予 "作为服务登录" 的特权的提示，dbsvc 也将会失败。</p>
-as	所有服务都用一个 Microsoft Windows 帐户运行。如果指定 -as ，服务会以 Microsoft Windows LocalSystem 帐户运行。不需要口令。 -a 和 -as 必须要使用一个。
-i	显示一个图标，双击它可以显示数据库服务器消息窗口。

创建选项	说明
-p	此选项与 -a 选项一起使用，用以指定服务运行时所用帐户的口令。
-rg <i>dependency,...</i>	指定一个或多个在所创建服务可以启动之前必须要启动的装载排序组。
-rs <i>dependency,...</i>	指定列表中的所有服务必须已经全部启动，才允许正在创建的服务启动。
-s <i>startup</i>	为 SQL Anywhere 服务设置启动行为。您可以将启动行为设置为 [Automatic]、[Manual] 或 [Disabled]。缺省值为 [Manual]。
-sd <i>description</i>	使用此选项提供服务的说明。该说明显示在 Windows 服务管理器中。
-sn <i>name</i>	<p>此选项用于为服务提供一个名称。该名称显示在 Windows 服务管理器中。如果不指定 -sn 选项，则缺省的服务名称为 SQL Anywhere - svc。例如，在缺省情况下，下面的服务命名为 SQL Anywhere - myserv。</p> <pre>dbsvc -as -w myserv "c:\Program Files\SQL Anywhere 11\bin32\dbeng11.exe"</pre> <p>为了使服务名称 myserv 显示在 Windows 服务管理器中，需要执行以下命令（所有内容应在一行中输入）：</p> <pre>dbsvc -as -sn myserv -w myserv "c:\Program Files\SQL Anywhere 11\bin32\dbeng11.exe"</pre>

创建选项	说明
-t <i>type</i>	<p>指定此服务的类型。您可以从以下类型中进行选择：</p> <ul style="list-style-type: none"> ● DBLTM SQL Anywhere 日志传送管理器 (LTM)。也可以将此服务类型指定为 LTM。请参见“日志传送管理器实用程序 (dbltm)”一节第 748 页。 ● dbmlsync MobiLink 同步客户端 (dbmlsync)。请参见“dbmlsync 语法”一节《MobiLink - 客户端管理》。 ● DBNS 广播转发器 (dbns11)。请参见“广播转发器实用程序 (dbns11)”一节第 703 页。 ● dblsn 监听器实用程序 (dblsn)。请参见“用于 Windows 设备的监听器实用程序”一节《MobiLink - 服务器启动的同步》。 ● DBRemote SQL Remote 消息代理 (dbremote)。请参见“消息代理 (dbremote)”一节《SQL Remote》。 ● MobiLink MobiLink 服务器 (mlsrv11)。请参见“mlsrv11 语法”一节《MobiLink - 服务器管理》。 ● 网络 SQL Anywhere 网络数据库服务器 (dbsrv11)。请参见“SQL Anywhere 数据库服务器”一节第 148 页。 ● rshost “中继服务器状态管理器”一节《MobiLink - 服务器管理》 ● RSOE “出站启动器”一节《MobiLink - 服务器管理》 ● Standalone SQL Anywhere 个人数据库服务器 (dbeng11)。请参见“SQL Anywhere 数据库服务器”一节第 148 页。 ● vss 卷影复制服务 (VSS)。请参见“使用 SQL Anywhere 卷影复制服务 (VSS)”一节第 829 页。 <p>所有服务类型的缺省设置为 [Standalone]。</p>

修饰符选项	说明
-cm	<p>显示用于创建服务的命令。此选项可用于将创建命令输出到一个文件中，随后可利用它在其它计算机上添加该项服务，或是在对服务做了更改时，用它来将服务恢复到其原始状态。在使用 -cm 时，必须指定 -g 选项或 -l 选项，否则命令将失败。指定 -g 会显示指定服务的创建命令，而指定 -l 会显示所有服务的创建命令。</p> <p>如果指定的服务不存在，则将生成用于删除服务的命令。例如，如果计算机上不存在 <code>service_1</code>，<code>dbsvc -cm -g service_1</code> 将会返回以下命令，以删除 <code>service_1</code> 服务：</p> <pre>dbsvc -y -d "service_1"</pre> <p>如果服务不使用 LocalSystem 帐户，由于无法获得口令，因此它没有包括在所生成的命令中。如果创建服务时使用了 <code>-a user -p password</code>，则输出中仅包括 <code>-a user</code>。</p>
-o log-file	<p>将服务实用程序 (dbsvc) 的输出写入指定的文件中。-o 选项必须位于 -d、-g、-l、-u 和 -x 选项之前。如果为 <code>dbsvc</code> 和作为服务运行的可执行文件（例如，数据库服务器）都指定 -o 选项，将会为两者都创建日志文件。例如：</p> <pre>dbsvc -o out1.txt -y -as -w mydsn install-dir\bin32\dbsrv11 -n mysrv -o c:\out2.txt</pre> <p>这种情况下，<code>dbsvc</code> 的输出记录到 <code>out1.txt</code> 中，而数据库服务器的输出则记录到 <code>c:\out2.txt</code> 中。</p>
-q	<p>在数据库服务器消息窗口中不显示消息。如果指定 -q，还建议您使用 -o 选项来指定记录消息的文件。如果您在修改或删除一个现有服务时指定此选项，则还必须指定 -y，否则操作将失败。</p>
-y	<p>自动执行操作，不提示您进行确认。此选项可以与 -w 或 -d 选项一起使用。如果您在修改或删除一个现有服务时指定 -q，则还必须指定 -y，否则操作将失败。</p>

注释

服务使用一组选项运行数据库服务器或其它应用程序。此实用程序为 Windows 上 SQL Anywhere 服务的管理提供了一套完整的方法。您必须是本地计算机上 [管理员] 组的成员，才能使用服务实用程序。

可以通过下列方式访问服务实用程序：

- 从 Sybase Central 使用 [创建服务向导]。请参见“创建 Windows 服务”一节第 60 页。
- 在命令提示符处，使用 `dbsvc` 命令。

退出代码是零（成功）或非零（失败）。请参见“软件组件的退出代码”一节《SQL Anywhere 服务器 - 编程》。

另请参见

- “了解 Windows 服务” 一节第 59 页

示例

创建名为 `myserv` 的个人服务器服务，它使用指定的参数启动指定的服务器。该服务器以 `LocalSystem` 用户身份运行：

```
dbsvc -as -w myserv "c:\Program Files\SQL Anywhere 11\bin32\dbeng11.exe" -n  
myeng -c 8m "c:\temp\mysample.db"
```

创建名为 `mynetworkserv` 的网络服务器服务。该服务器使用本地帐户运行，并在重新启动计算机时自动启动：

```
dbsvc -as -s auto -t network -w mynetworkserv "c:\Program Files\SQL Anywhere  
11\bin32\dbsrv11.exe"  
-x tcpip -c 8m "c:\temp\mysample.db"
```

列出关于 `myserv` 服务的所有详细信息：

```
dbsvc -g myserv
```

删除名为 `myserv` 的服务，但不提示您进行确认：

```
dbsvc -y -d myserv
```

创建依赖于 `Workstation` 服务和 `TDI` 组的服务：

```
dbsvc -rs lanmanworkstation -rg TDI -w ...
```

创建名为 `mysyncservice` 的服务：

```
dbsvc -as -s manual -t dbmlsync -w mysyncservice "c:\Program Files\SQL  
Anywhere 11\bin32\dbmlsync.exe"  
-c "SQL Anywhere 11 CustDB"
```

生成用于创建服务 `service_1` 的命令，并将其输出到名为 `restoreservice.bat` 的文件中：

```
dbsvc -cm -g service_1 > restoreservice.bat
```

`restoreservice.bat` 文件包含以下内容：

```
dbsvc -t Standalone -s Manual -as -y -w "service_1"  
"c:\Program Files\SQL Anywhere 11\bin32\dbeng11.exe"
```

创建可手工启动的 `MobiLink` 监听器服务：

```
dbsvc -as -i -w myListener "c:\Program Files\SQL Anywhere 11\bin32\dblsn.exe"  
"@c:\temp\dblsn.opt"
```

启动 `myListener` 服务：

```
dbsvc -u myListener
```

停止 `myListener` 服务：

```
dbsvc -x myListener
```

创建在数据库服务器启动时自动启动的卷影复制服务 (VSS)：

```
dbsvc -as -s Automatic -t vss -w SAVSSWriter "c:\Program Files\SQL Anywhere  
11\bin32\dbvss11.exe"
```

SQL Anywhere 控制台实用程序 (dbconsole)

为数据库服务器连接提供管理和监控工具。

语法

dbconsole [*options*]

选项	说明
@ <i>data</i>	此选项用于从指定的环境变量或配置文件中读入选项。请参见“ 使用配置文件 ”一节第 695 页。 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“ 文件隐藏实用程序 (dbfhide) ”一节第 724 页。
-c " <i>keyword=value; ...</i> "	指定连接参数。请参见“ 连接参数 ”一节第 248 页。
-datasource <i>DSN-name</i>	指定要连接的 ODBC 数据源。不使用 iAnywhere JDBC 驱动程序时也可使用此选项。
-host <i>hostname</i>	指定正运行数据库服务器的计算机的 <i>hostname</i> 或 IP 地址。可以使用名称 localhost 来代表当前计算机。
-port <i>port-number</i>	指定数据库服务器使用的端口号。SQL Anywhere 的缺省端口号是 2638 。

注释

SQL Anywhere 控制台允许从客户计算机监控服务器。此实用程序也称为网络服务器监控器。您可以使用它跟踪登录到网络上其它位置的数据库服务器的用户。您还可以在本地客户端屏幕上显示服务器和客户端统计、断开用户的连接和配置数据库服务器。SQL Anywhere 控制台可以显示多个连接的信息。

◆ 要断开用户与数据库的连接

1. 从 SQL Anywhere 控制台连接到数据库。
2. 在 [用户 ID] 列中，右击用户并选择 [断开连接]。

在 [选项] 窗口（可通过选择 [文件] » [选项] 进行访问）中可以配置 SQL Anywhere 控制台中显示的列。请参见“[使用 SQL Anywhere 控制台实用程序](#)”一节第 687 页。

SQL Anywhere 控制台可以在除 Windows Mobile、AIX、HP-UX 和 HP-UX Itanium 外的所有支持的平台上使用。在这些平台上，可以使用连接级、服务器级和数据库级属性来获取信息，也可以从一台运行支持 SQL Anywhere 控制台的操作系统（如 Windows、Mac OS X 或 Linux）的计算机上监控服务器。

有关获取属性值的详细信息，请参见“[连接属性、数据库属性和数据库服务器属性](#)”第 565 页。

后台启动服务器实用程序 (dbspawn)

在后台启动数据库服务器。

语法

dbspawn [*options*] *server-command*

选项	说明
@data	从指定的环境变量或配置文件中读入选项。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。后台启动服务器实用程序 (dbspawn) 不会展开由 @data 选项指定的配置文件的内容。请参见“ 使用配置文件 ”一节第 695 页。 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“ 文件隐藏实用程序 (dbfhide) ”一节第 724 页。
-f	强制 dbspawn 启动数据库服务器，即使已存在缺省数据库服务器。如果正在运行的数据库服务器不是缺省服务器，dbspawn 会启动其它服务器。 如果已经有一个与 dbspawn 所尝试启动的数据库服务器同名的数据库服务器正在运行，dbspawn 会返回成功但不启动新服务器。
-p	指定数据库服务器进程的操作系统进程 ID。例如： <pre>dbspawn -p dbeng11 -n newserver</pre> 在命令提示符处报告以下形式的消息： <pre>New process ID is 306</pre>
-q	以安静模式运行—不显示消息。
<i>server-command</i>	指定启动数据库服务器的命令行。请参见“ SQL Anywhere 数据库服务器 ”一节第 148 页。

注释

dbspawn 实用程序用于在后台启动服务器。dbspawn 在后台启动服务器，并带着退出代码零（成功）或非零（失败）返回。如果在同一计算机上已经运行了一个数据库服务器，dbspawn 不会启动新服务器，而是报告操作失败。否则，dbspawn 会在数据库服务器完成初始化并准备好接受请求之后返回。

有关退出代码的详细信息，请参见“[软件组件的退出代码](#)”一节《[SQL Anywhere 服务器 - 编程](#)》。

dbspawn 实用程序对于从批处理文件启动服务器非常有用，尤其是在批处理文件中的后续命令需要正在接受请求的服务器时。

如果指定的路径中包含至少一个空格，则必须用一对双引号将路径括起。例如，

```
dbspawn dbeng11 "c:\my databases\mysalesdata.db"
```

如果指定的路径中不包含空格，则不需要使用引号。

停止服务器实用程序 (dbstop)

停止数据库或数据库服务器。

语法

dbstop [*options*] [*server-name*]

选项	说明
<i>@data</i>	从指定的环境变量或配置文件中读入选项。请参见“使用配置文件”一节第 695 页。 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“文件隐藏实用程序 (dbfhide)”一节第 724 页。
-c <i>"keyword=value; ..."</i>	指定连接字符串。要停止网络服务器，连接字符串必须包含拥有服务器停止权限的用户 ID。缺省情况下，在网络服务器上需要有 DBA 权限，所有用户都可以关闭个人服务器，但可以使用 -gk 服务器选项对此进行修改。 如果提供连接参数，则不要再提供服务器名。请参见“连接参数”一节第 248 页、“Unconditional 连接参数 [UNC]”一节第 282 页和“ -gk 服务器选项”一节第 181 页。
-d	不停止数据库服务器，而是只停止在连接字符串中指定的数据库。
-o <i>filename</i>	将输出消息写入指定的文件中。
-q	以安静模式运行—不显示消息。
-x	如果仍有到服务器的活动连接，则不停止服务器。指定此选项可以阻止 dbstop 在存在活动连接时提示用户进行确认。
-y	停止服务器，即使仍有到服务器的活动连接。这与在连接参数中包含 Unconditional=YES 具有相同的效果。
<i>server-name</i>	指定当前计算机所运行数据库服务器的名称。必须这样启动数据库服务器：在关闭它时不需要权限。缺省情况下，个人数据库服务器以此模式启动。对于网络数据库服务器，您必须提供 -gk all 选项。请参见“ -gk 服务器选项”一节第 181 页。 如果提供服务器名，则不要再提供连接参数。

注释

停止服务器实用程序用于使数据库服务器停止运行。可以使用 **-d** 选项停止指定的数据库。

停止服务器实用程序只能在命令提示符环境中运行。在窗口式环境中，您可以通过单击数据库服务器消息窗口中的 [关闭] 来停止数据库服务器。

选项可让您控制是否停止服务器（即使有活动连接）；以及控制是停止服务器还是只停止数据库。

在服务器上有活动连接的情况下，`dbstop` 的行为是可以控制的。如果具有活动连接，则 `dbstop` 会给出一个提示，询问您是否希望关闭服务器。可以使用 `-x` 和 `-y` 选项来更改这一行为。

如果 `dbstop` 能够停止数据库服务器，则只有在所有数据库停止运行并且数据库服务器也停止后，`dbstop` 才会结束。（数据库服务器完全停止后，另一个服务器才能以相同的名称和数据库启动。）`dbstop` 成功完成后，数据库服务器进程可能仍在运行，其某些资源（例如由 `-o` 服务器选项指定的输出文件）可能仍在运行。

退出代码是零（成功）或非零（失败）。请参见“[软件组件的退出代码](#)”一节《[SQL Anywhere 服务器 - 编程](#)》。

如果在定义了 `SQLCONNECT` 环境变量的环境中使用 `dbstop`，应指定 `-c` 选项。否则，可能会出现意外的结果。

示例

正在运行名为 `myserver` 的服务器，该服务器没有启动数据库。若要停止该服务器，请指定实用程序数据库作为 `DatabaseName (DBN)` 连接参数：

```
dbstop -c "UID=DBA;PWD=sql;ENG=myserver;DBN=utility_db"
```

正在运行名为 `myserver` 的服务器，并且已启动了 `demo.db` 数据库。要停止该服务器和数据库：

```
dbstop -c "UID=DBA;PWD=sql;ENG=myserver"
```

正在运行名为 `myserver` 的个人服务器。要停止该服务器和数据库而不管是否存在连接：

```
dbstop -y myserver
```

正在运行名为 `myserver` 的服务器，并且已启动了 `demo.db` 数据库。要仅停止名为 `demo` 的数据库，但不停止其它数据库或服务器自身，请执行以下命令：

```
dbstop -c "UID=DBA;PWD=sql;ENG=myserver;DBN=demo" -d
```

支持实用程序 (dbsupport)

将有关错误和软件使用方面的信息发送给 iAnywhere Solutions。

语法

dbsupport [*options*] *operation* [*operation-specific-option*]

dbsupport *configuration-options*

选项	说明
@ <i>data</i>	从指定的环境变量或配置文件中读入选项。请参见“使用配置文件”一节第 695 页。 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“文件隐藏实用程序 (dbfhide)”一节第 724 页。
-o <i>filename</i>	将输出发送到指定文件中。
-q	只显示重要消息。

操作	说明
-e <i>configuration-option</i>	显示指定配置选项的设置。例如，假设您运行以下命令来配置 dbsupport 在可能的情况下显示提示： <code>dbsupport -cc promptdefy</code> 运行命令 <code>dbsupport -ecc</code> 时，会返回以下设置： <code>-cc "promptdefy"</code>
-is <i>submission-ID</i> [-rr <i>N</i>]	检查已提交给 iAnywhere Solutions 的崩溃报告的状态。 例如，以下命令将查询 ID 66 的提交状态： <code>dbsupport -is 66</code>
-iu [-r <i>N</i>]	检查您所用的 SQL Anywhere 版本是否有更新。 也可以使用 Interactive SQL 和 Sybase Central 来检查更新。请参见“检查软件更新”一节第 690 页。
-lc	生成一份所有尚未提交给 iAnywhere Solutions 的崩溃报告的列表。所列出的报告名称可以与 -sc 选项一起使用。

操作	说明
-ls	<p>生成所有已提交给 iAnywhere Solutions 的报告的提交 ID 列表。例如：</p> <pre>dbsupport -ls</pre> <p>其返回的信息如下：</p> <pre>Submission ID: 4 Minicore dump 20051220_133828_32116 reported: Wed Mar 15 16:31:56 2006 Submission ID: 98 Minicore dump 20051229_221211_3221 reported: Wed Mar 22 16:33:26 2006</pre>
-pc filename	显示崩溃报告信息。通过此选项，可以在将信息提交给 iAnywhere Solutions 之前进行查看。
-pd	显示已收集的诊断信息。通过此选项，可以在将信息提交给 iAnywhere Solutions 之前进行查看。
-ps submission-ID	<p>显示已提交给 iAnywhere Solutions 的特定报告的信息。例如：</p> <pre>dbsupport -ps 4</pre> <p>有关提交 4 的信息将被返回：</p> <pre>Minicore dump 20051220_133828_32116 reported: Wed Mar 15 16:31:56 2006</pre>
-sa [-r number-of-submission-retries]	将所有存储于诊断目录中的崩溃报告和诊断信息提交给 iAnywhere Solutions。
-sc reportname [-r number-of-submission-retries] [-nr -rr N]	<p>将一份崩溃报告和诊断信息提交给 iAnywhere Solutions。例如：</p> <pre>dbsupport -sc 20051220_133828_32116</pre> <p>使用 -lc 选项可以查看未提交报告的列表。</p>
-sd [-r number-of-submission-retries]	<p>仅将诊断信息提交给 iAnywhere Solutions。</p> <p>有关诊断目录的详细信息，请参见“SADIAGDIR 环境变量”一节第 353 页。</p>

配置选项	说明
-cc [autosubmit no promptDefY promptDefN]	<p>更改 dbsupport 的提示行为。您可以指定以下各选项之一：</p> <ul style="list-style-type: none"> ● autosubmit 自动提交报告。 ● no 不提示用户授权来提交报告。报告和功能统计信息将不会提交。 ● promptDefY 在可能的情况下提示用户授权来提交报告。如果用户没有给予回答，则提交报告。 ● promptDefN 在可能的情况下提示用户授权来提交报告。如果用户没有给予回答，则不提交报告。这是缺省行为。 <p>例如，如果正在应用程序中使用嵌入式 SQL Anywhere，则您可能希望将支持实用程序配置为不向 iAnywhere Solutions 提交报告。</p> <p>如果指定此选项，其值将成为支持实用程序所用的缺省值。该设置存储于诊断目录的 <i>dbsupport.ini</i> 文件中。</p> <p>以下命令将支持实用程序配置为不提交报告并且不提示用户提交报告：</p> <pre>dbsupport -cc no</pre>
-cd <i>retry-delay</i>	<p>指定自上次尝试失败后报告提交操作的重试延迟（以秒为单位）。缺省延迟为 30 秒。</p> <p>如果指定此选项，其值将成为支持实用程序所用的缺省值。该设置存储于诊断目录的 <i>dbsupport.ini</i> 文件中。</p> <p>以下 dbsupport 命令指定每 3 秒将失败的提交重试一次，在放弃重试前重试次数可达 4 次：</p> <pre>dbsupport -cr 4 -cd 3</pre>
-ce <i>email-address;email-server[:port][;user-id:password]</i>	<p>指定在出现崩溃后发出电子邮件的地址。将使用 <i>email-server</i> SMTP 服务器发送电子邮件。或者，可以指定要使用的端口以及用于 SMTP 服务器验证的 <i>user-id</i> 和 <i>password</i>。</p>
-cet	<p>测试由 -ce 选项指定的电子邮件设置。</p>

配置选项	说明
<p>-ch <i>crash-handler-program</i></p>	<p>指定在出现崩溃时所调用的程序。</p> <p>如果指定此选项，其值将成为支持实用程序所用的缺省值。该设置存储于诊断目录的 <i>dbsupport.ini</i> 文件中。</p> <p>数据库服务器支持以下三种替代参数，用以设置传递给 <i>crash-handler-program</i> 的信息：</p> <ul style="list-style-type: none"> ● %F 此参数将被所生成报告文件的位置的完整路径代替。 ● %P 此参数将被生成报告的程序的名称所代替。例如，如果报告由版本 11 个人数据库服务器生成，则将返回 <i>dbeng11</i>。 ● %S 此参数将被出现崩溃或致命错误时正在使用的数据库服务器的名称所代替。例如，如果报告由名为 <i>Sample</i> 的数据库服务器生成，则将返回 <i>Sample</i>。 <p>可以使用 <i>\$F</i>、<i>\$P</i> 和 <i>\$S</i> 来代替 <i>%F</i>、<i>%P</i> 和 <i>%S</i>。由于不同的命令解释程序会对字符 <i>%</i> 和 <i>\$</i> 进行不同解释，因此提供了这两种形式。例如，<i>%F</i> 在 4NT 上会被替换为环境变量 <i>F</i> 的值；可以使用 <i>\$F</i> 来避免这种替换。</p> <p>假设在文件 <i>c:\test.bat</i> 中有一个崩溃处理程序，其包含以下命令：</p> <pre>copy %1 c:\archives echo %2</pre> <p>在 Windows 中，以下命令告诉 <i>dbsupport</i> 在出现崩溃时使用两个参数来启动 <i>c:\test.bat</i>。如果此报告正在提交中，会在提交报告前调用此程序。</p> <pre>dbsupport -ch "c:\test.bat \"%F\" parm2"</pre> <p>由 <i>%F</i> 所指定的替代路径作为第一个参数发送给 <i>c:\test.bat</i>。参数 <i>parm2</i> 作为第二个参数发送给 <i>c:\test.bat</i>。注意，必须使用引号来指定将接受参数的崩溃处理程序。</p> <p>在上例中，在所生成报告文件的完整路径两侧使用了附加的引号。为避免在访问 <i>dbsupport</i> 正使用的报告文件时出现问题，崩溃处理程序会创建其自身的报告文件副本。</p>

配置选项	说明
-ch-	<p>删除存储于 <i>dbsupport.ini</i> 文件中的崩溃处理程序设置。例如：</p> <pre>dbsupport -ch-</pre>
-cid <i>customer-id</i>	<p>指定一个用于在提交报告中标识您身份的字符串。如果指定此选项，其值将成为支持实用程序所用的缺省值。该配置存储于诊断目录的 <i>dbsupport.ini</i> 文件中。</p> <p>以下示例为 dbsupport 指定一个客户标识字符串：</p> <pre>dbsupport -cid myid@company.com dbsupport -cid "MyClientApp 1.0"</pre>
-cid-	<p>从 <i>dbsupport.ini</i> 文件中删除客户标识字符串。例如：</p> <pre>dbsupport -cid-</pre>
-cp { <i>email-server</i> [<i>:port</i>] autodetect }	<p>配置在提交错误报告时所用的 HTTP 代理主机和端口。</p> <p>在 Windows 上，还支持语法 -cp autodetect。指定此选项后，如果已经使用 Internet Explorer 设置了代理服务器和端口而且它们当前已经启用，dbsupport 会使用该系统设置来配置其代理服务器和端口。可以在 Windows 的 Internet Explorer 中设置代理服务器和端口，方法是：选择 [工具] » [选项] » [局域网设置]。</p>
-cp-	<p>从 <i>dbsupport.ini</i> 文件中删除 HTTP 代理主机和端口设置。例如：</p> <pre>dbsupport -cp-</pre>
-cr <i>number-of-submission-retries</i>	<p>指定对于失败的提交操作的重试次数。</p> <p>如果指定此选项，其值将成为支持实用程序所用的缺省值。该设置存储于诊断目录的 <i>dbsupport.ini</i> 文件中。</p> <p>以下 dbsupport 命令指定每 3 秒将失败的操作重试一次，在放弃重试前重试次数可达 4 次：</p> <pre>dbsupport -cr 4 -cd 3</pre>

特定于操作的选项	说明
-nr	指定 <code>dbsupport</code> 不在服务器中检查提交状态。例如，以下命令将提交报告，但不检查这一新提交的状态： <code>dbsupport -nr -sc 20051220_133828_32116</code> 缺省情况下， <code>dbsupport</code> 会检查针对所提交的问题是否已经有解决方案。
-r <i>number-of-submission-retries</i>	指定 <code>dbsupport</code> 将尝试发送提交的最大次数。指定 0 表示重试次数无限制。缺省值为 10。指定 <code>-r</code> 时，将替换存储于 <code>dbsupport.ini</code> 中的 <code>-cr</code> 值（如果存在）。
-rd <i>retry-delay</i>	指定 <code>dbsupport</code> 在两次报告重发尝试之间等待的秒数。缺省值为 30。指定 <code>-rd</code> 时，将替换存储于 <code>dbsupport.ini</code> 文件中的 <code>-cd</code> 值（如果存在）。
-rr <i>number-of-submission-response-retries</i>	指定 <code>dbsupport</code> 将尝试获取提交响应的最大次数。指定 0 表示重试次数无限制。缺省值是 10。

注释

支持实用程序 (`dbsupport`) 可用于以下任何任务：

- 通过 Internet 将诊断信息和崩溃报告提交给 iAnywhere Solutions
- 提交功能统计信息
- 列出有关已提交和未提交崩溃报告的信息
- 打印有关已提交和未提交崩溃报告的信息
- 查询提交状态
- 查询针对您所用版本的 SQL Anywhere 是否有可用的软件更新
- 配置当数据库或 MobiLink 服务器检测到致命错误（声明/崩溃）时所要执行的操作

缺省情况下，`dbsupport` 会检查针对所提交的问题是否已经有解决方案。

可以在发生致命错误时，将来自以下任何应用程序的信息以错误报告的形式发送出去：

- Interactive SQL (dbisql)
- MobiLink 监听器 (dblsn)
- MobiLink 服务器 (mlsrv)
- 网络服务器 (dbsrv11)
- 个人服务器 (dbeng11)
- QAnywhere 代理 (qaagent)
- 复制代理 (dbltn)
- 用于 MobiLink 的 SQL Anywhere 客户端 (dbmlsync)
- SQL Anywhere 控制台实用程序 (dbconsole)
- SQL Remote (dbremote)
- Sybase Central

报告成功提交后，会获得一个唯一的提交 ID。各项报告均写入诊断目录中。

平台	缺省诊断目录位置
Windows (Windows Mobile 除外)	%ALLUSERSPROFILE%\Application Data\SQL Anywhere 11\diagnostics
Windows Mobile	运行可执行文件的目录。
Unix	\$HOME/.sqlanywhere11/diagnostics

有关诊断目录的信息，请参见“[SADIAGDIR 环境变量](#)”一节第 353 页。

有关错误报告及其提交方式的详细信息，请参见“[SQL Anywhere 中的错误报告](#)”一节第 76 页。

支持实用程序也可配置成在检测到问题时执行某些操作。例如，可将它配置为在每次数据库服务器提交错误报告时执行指定的处理程序。要向错误处理程序添加自定义的操作时，此功能非常有用。

支持实用程序经配置也可以重试某些操作。例如，经过配置，它可以在提交报告时在 30 秒后再次重试这一操作，重试次数可达 10 次。在处理服务有可能会暂时无法使用的情况时，此功能非常有用。

支持实用程序的设置存储在诊断目录的 *dbsupport.ini* 文件中。

在替换缺省行为（包括存储于 *dbsupport.ini* 文件中的行为）时，特定于操作的选项非常有用。

另请参见

- “[SADIAGDIR 环境变量](#)”一节第 353 页
- “[软件组件的退出代码](#)”一节 《[SQL Anywhere 服务器 - 编程](#)》

事务日志实用程序 (dblog)

管理数据库事务日志。

语法

dblog [*options*] *database-file*

选项	说明
@data	从指定的环境变量或配置文件中读入选项。请参见“使用配置文件”一节第 695 页。 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“文件隐藏实用程序 (dbfhide)”一节第 724 页。
-ek key	直接在命令行中为高度加密的数据库指定加密密钥。如果您有一个高度加密的数据库，则必须提供加密密钥，才能在任何情况下使用数据库或事务日志。对于高度加密数据库，您必须指定 -ek 或 -ep ，但不要同时指定这两者。如果您没有为高度加密数据库指定正确的密钥，则该命令将失败。
-ep	指定需要系统提示输入加密密钥。使用此选项将显示一个窗口，可以在其中输入加密密钥。这样，加密密钥决不会以明文显示，从而提供了额外的安全保证。对于高度加密数据库，您必须指定 -ek 或 -ep ，但不要同时指定这两者。如果您没有为高度加密数据库指定正确的密钥，则该命令将失败。
-g n	如果使用日志传送管理器参与复制服务器安装，则使用此选项。此选项可用于在恢复备份之后设置世代号。它执行与以下复制服务器函数相同的功能： <pre>dbcc settrunc('ltm', 'gen_id', n)</pre> 有关世代号和 dbcc 的信息，请参见复制服务器文档。
-il	如果您已经停止使用日志传送管理器参与该数据库上的复制服务器安装，但继续使用 SQL Remote 或 MobiLink 同步，则可以使用此选项。它会重置用于 <code>delete_old_logs</code> 选项的日志传送管理器日志偏移，以允许将不再需要的事务日志删除。 它执行与以下复制服务器函数相同的函数： <pre>dbcc settrunc('ltm', 'ignore')</pre> 有关 dbcc 的信息，请参见复制服务器的文档。
-ir	如果您已经停止在此数据库上使用 SQL Remote，但继续使用日志传送管理器或 MobiLink 同步，则可以使用此选项。它重置为 <code>delete_old_logs</code> 选项保留的 SQL Remote 日志偏移，允许将不再需要的事务日志删除。
-is	如果您已经停止在此数据库上使用 MobiLink 同步，但继续使用日志传送管理器或 SQL Remote，则可以使用此选项。它重置为 <code>delete_old_logs</code> 选项保留的 MobiLink 日志偏移，允许将不再需要的事务日志删除。

选项	说明
-m <i>mirror-name</i>	为新的事务日志镜像指定文件名。如果数据库当前没有使用事务日志镜像，则它会开始使用一个。如果数据库已在使用事务日志镜像，则它会改为将新文件用作它的事务日志镜像。
-n	停止使用事务日志，并停止使用事务日志镜像。如果没有事务日志，则数据库不能再参与数据复制，也不能在数据恢复时再使用事务日志。如果存在 SQL Remote、日志传送管理器或 dbmsync 截断偏移，则不能删除事务日志，除非同时指定了相应的忽略选项（对应于日志传送管理器的 -il、对应于 SQL Remote 的 -ir 或对应于 dbmsync 的 -is）。如果数据库已经打开审计功能，则不能停止使用事务日志（除非先将审计功能关闭）。
-o <i>filename</i>	将输出消息写入指定的文件中。
-q	以安静模式运行—不显示消息。
-r	为维护事务日志镜像的数据库维护单个事务日志。
-t <i>log-name</i>	为新的事务日志指定文件名。如果数据库当前没有使用事务日志，它会开始使用一个。如果数据库已在使用事务日志，则它会改为将新文件用作其事务日志。
-x <i>n</i>	将事务日志当前相对偏移重置为 <i>n</i> ，使数据库能够参与复制。此选项用于重装 SQL Remote 统一数据库。请参见“ 将远程数据库抽取到重装文件 ”一节《 SQL Remote 》。
-z <i>n</i>	将事务日志起始偏移重置为 <i>n</i> ，使数据库能够参与复制。此选项用于重装 SQL Remote 统一数据库。请参见“ 将远程数据库抽取到重装文件 ”一节《 SQL Remote 》。

注释

通过 dblog 实用程序，您可以显示或更改与某数据库关联的事务日志或事务日志镜像的名称，也可以使数据库停止维护事务日志或镜像，或者开始维护事务日志或镜像。

事务日志镜像是事务日志的副本，由数据库同时维护。

事务日志的名称最初是在数据库初始化时设置的。事务日志实用程序处理数据库文件。更改事务日志文件名时，数据库服务器不能正在运行该数据库，否则会出现错误消息。

此实用程序显示有关事务日志的其它信息，其中包括：

- 版本号
- 事务日志文件的名称
- 事务日志镜像文件的名称（如果有）
- 当前相对偏移

您可以以下列方式访问事务日志实用程序：

- 从 Sybase Central 使用 [更改日志文件设置向导]。请参见“更改事务日志的位置”一节第 15 页。
- 从 Interactive SQL 使用 ALTER DATABASE *dbfile* ALTER LOG 语句。请参见“ALTER DATABASE 语句”一节《SQL Anywhere 服务器 - SQL 参考》。
- 在命令提示符处，使用 dblog 命令。

退出代码是零（成功）或非零（失败）。请参见“软件组件的退出代码”一节《SQL Anywhere 服务器 - 编程》。

卸载实用程序 (dbunload)

将数据库卸载到 SQL 命令文件中。

语法

dbunload [*options*] [*directory*]

选项	说明
@ <i>data</i>	<p>从指定的环境变量或配置文件中读入选项。请参见“使用配置文件”一节第 695 页。</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“文件隐藏实用程序 (dbfhide)”一节第 724 页。</p>
-ac " <i>keyword=value; ...</i> "	<p>连接现有数据库并将数据重装到其中，它将卸载某个数据库并将结果重装到一个现有数据库中的操作结合起来。在 Windows Mobile 中不支持此选项。</p> <p>例如，可以使用初始化实用程序创建一个新数据库，然后使用此选项进行重装。在希望更改初始化选项时，这一方法非常有用。</p> <p>以下命令（所有内容应在一行中输入）将 <i>c:\mydata.db</i> 数据库的副本装载到名为 <i>c:\mynewdata.db</i> 的现有数据库文件中：</p> <pre>dbunload -c "UID=DBA;PWD=sql;DBF=c:\mydata.db" -ac "UID=DBA;PWD=sql;DBF=c:\mynewdata.db"</pre> <p>如果原始数据库是使用 SQL Anywhere 版本 9 或更早版本创建的，并且新数据库尚未运行，则必须在 -ac 选项中提供数据库服务器的名称。例如：</p> <pre>dbunload -c "UID=DBA;PWD=sql;DBF=c:\mydata.db" -ac "UID=DBA;PWD=sql;DBF=c:\mynewdata.db;ENG=newserver"</pre> <p>如果使用此选项，则不会在磁盘上创建数据的中间副本，因而无需在命令行中指定卸载目录。这为数据提供了更高的安全性。</p>

选项	说明
<p>-an database</p>	<p>使用此选项可将卸载数据库、创建新数据库和装载数据这三个操作组合起来。此选项在 Windows Mobile 上不受支持，在 Intel 的 Mac OS X 上重建版本 9 或更早版本的数据库时也不受支持。</p> <p>在创建源数据库时所指定的选项将用于创建新的数据库。但是，可以根据需要通过指定其它受支持的 dbunload 选项来更改初始化选项（例如指定 -ap 以更改页面大小或指定 -et 以启用表加密）。</p> <p>例如，以下命令（所有内容应在一行中输入）将创建名为 <i>mydatacopy.db</i> 的新数据库文件，并将 <i>mydata.db</i> 的模式和数据复制到其中：</p> <pre>dbunload -c "UID=DBA;PWD=sql;DBF=c:\mydata.db" -an c:\mydatacopy.db</pre> <p>如果使用此选项，则不会在磁盘上创建数据的中间副本，因而无需在命令行中指定卸载目录。这为数据提供了更高的安全性。</p> <p>在创建新数据库时，如果新数据库的 dbspace 文件与原数据库的 dbspace 文件位于同一目录中，新的 dbspace 文件名的末尾会附加一个 R 字母，以避免文件名冲突。例如，如果所卸载数据库在文件 <i>library.db</i> 中的 dbspace 称为 <i>library</i>，则新数据库的 <i>library</i> dbspace 为 <i>library.dbR</i>。</p> <p>-an 所指定的文件相对于数据库服务器。</p>
<p>-ap size</p>	<p>设置新数据库的页面大小。除非另外指定了 -an 或 -ar，否则将忽略此选项。数据库的页面大小（以字节为单位）可以是 2048、4096、8192、16384 或 32768，缺省值为原数据库的页面大小。必须使用该选项指定 -an 或 -ar。如果数据库服务器上已有数据库在运行，则服务器的页面大小（使用 -gp 选项设置）必须大到足以处理新的页面大小。请参见“-gp 服务器选项”一节第 183 页。</p>

选项	说明
<p>-ar [<i>directory</i>]</p>	<p>以旧数据库的设置创建新数据库，重装新数据库，并替换旧数据库。但是，可以根据需要通过指定其它受支持的 dbunload 选项来更改初始化选项（例如指定 -ap 以更改页面大小或指定 -et 以启用表加密）。</p> <p>如果您使用此选项，则不能有到该数据库的其它连接，且数据库连接必须是本地的，而不是在网络上。此选项在 Windows Mobile 上不受支持，在 Intel 的 Mac OS X 上重建版本 9 或更早版本的数据库时也不受支持。</p> <p>如果指定可选的 <i>directory</i>，则事务日志偏移会因复制目的而被重置，来自旧数据库的事务日志会被移动到指定的目录中。指定的目录应该是保存由 [消息代理] 和 [复制代理] 使用的旧事务日志的目录。只有在数据库参与到复制中时，才会进行事务日志管理：如果没有 SQL Remote 发布者或 LTM 检查，则不需要旧事务日志，旧事务日志会被删除，而不会复制到指定的目录中。请参见“备份参与同步和复制的数据库”一节第 847 页。</p> <p>在创建新数据库时，如果新数据库的 dbspace 文件与原数据库的 dbspace 文件位于同一目录中，新的 dbspace 文件名的末尾会附加一个 R 字母，以避免文件名冲突。例如，如果所卸载数据库在文件 <i>library.db</i> 中的 dbspace 称为 library，则新数据库的 library dbspace 为 <i>library.dbR</i>。</p> <p>在重建加密的数据库时，用于原数据库和新数据库的加密密钥必须相同。</p> <p>使用 -ar 选项将数据库截断点重置为零。</p>
<p>-c "keyword=value; ..."</p>	<p>指定源数据库的连接参数。有关连接参数的说明，请参见“连接参数”一节第 248 页。用户 ID 应具有 DBA 权限以确保用户有权访问数据库中的所有表。</p> <p>例如，以下语句将卸载示例数据库，它会以用户 ID DBA 和口令 sql 进行连接。这些数据被卸载到 <i>c:\unload</i> 目录中。</p> <pre>dbunload -c "DBF=samples-dir\demo.db;UID=DBA;PWD=sql" c:\unload</pre> <p>有关 <i>samples-dir</i> 的信息，请参见“示例目录”一节第 368 页。</p>

选项	说明
-cm { sql dbinit }	<p>在服务器消息窗口中显示 CREATE DATABASE 或 dbinit 命令，以创建一个与正在卸载的数据库相同的数据库。如果另外还指定了 -an，则显示的命令是用于创建新数据库的命令。</p> <ul style="list-style-type: none"> ● sql 显示写入到 <i>reload.sql</i> 文件的 CREATE DATABASE 语句。 ● dbinit 显示初始化实用程序 (dbinit) 命令。 <p>针对现有的强加密数据库显示该语句或命令时（未指定 -an），无法从数据库获得加密密钥，因此，将在 ENCRYPTED 子句或 -ek 选项中出现一个问号 (?)。</p> <p>如果卸载一个使用版本 10 或更早版本数据库服务器创建的数据库，则不会显示创建命令或语句。</p>
-cp	<p>通过将 COMPRESSED 关键字附加到它所执行的 UNLOAD TABLE 语句中来压缩表数据输出文件。与 -an 或 -ar 选项一同指定时，此选项无效。</p>
-d	<p>如果使用此选项，将不会生成任何数据库定义命令（CREATE TABLE、CREATE INDEX，等等）；<i>reload.sql</i> 中只包含用于重装数据的语句。</p>
-dc	<p>强制重新计算数据库中所有计算列。缺省情况下，计算列的值不会被重新计算。指定 -dc 选项后，<i>reload.sql</i> 脚本中会加入一个新的代码段，用以对计算列进行重新计算。添加的语句形式如下。</p> <pre>ALTER TABLE "owner"."table-name" ALTER "computed-column" SET COMPUTE (compute-expression);</pre> <p>如果表中包含上下文相关的计算值（例如 CURRENT DATE），建议使用 ALTER TABLE 语句来对计算列的值进行重新计算，而不应使用 -dc 选项。请参见“ALTER TABLE 语句”一节《SQL Anywhere 服务器 - SQL 参考》。</p>
-e table, ...	<p>从 <i>reload.sql</i> 文件中排除指定的表。表名始终不区分大小写，即使在区分大小写的数据库中也是如此。</p> <p>使用 -e 选项创建的 <i>reload.sql</i> 文件不应用于数据库的重建，因为该文件并不包含所有的数据库表。如果有外键引用某个表，则必须有该表的内容才能重建数据库。</p> <p>建议只将 -e 选项和 -d 选项一起使用，用以卸载除 -e 所指定表外的所有表的数据。</p>

选项	说明
<p>-ea algorithm</p>	<p>指定用于数据库或表加密 (-et) 的加密算法。指定 -ea simple 可进行简单加密（不要指定 -ek 或 -ep）。简单加密等效于模糊处理，其目的只是为了使数据处于隐藏状态，以防有人意外地直接访问数据库文件，从而使要破译数据库中数据的人难于使用磁盘实用程序查看文件。</p> <p>为获得更高的安全性，可指定 AES 或 AES256 以分别实现 128 位或 256 位高度加密。指定 AES_FIPS 或 AES256_FIPS 以分别实现 128 位或 256 位 FIPS 认可的加密。对于高度加密，还必须指定 -ek 或 -ep 选项。有关高度加密的详细信息，请参见“高度加密”一节第 1019 页。</p> <p>要创建未加密的数据库，请指定 -ea none 或者不包含 -ea 选项（并且不要指定 -e、-et、-ep 或 -ek）。</p> <p>如果未指定 -ea 选项，则缺省行为如下：</p> <ul style="list-style-type: none"> ● 如果未指定 -ek、-ep 或 -et，则为 -ea none ● 如果指定了 -ek 或 -ep（带或不带 -et），则为 -ea AES ● 如果指定了 -et，而未指定 -ek 或 -ep，则为 -ea simple <p>算法名称是不区分大小写的。</p> <div style="border: 1px solid black; padding: 5px; margin-top: 10px;"> <p>需要单独授予许可的组成部分</p> <p>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。</p> <p>请参见“单独授权的组件”一节《SQL Anywhere 11 - 简介》。</p> </div>
<p>-ek key</p>	<p>如果卸载并重装数据库（使用 -an 选项），则可以使用此选项在 dbunload 命令中为新建的数据库指定加密密钥。如果您创建高度加密的数据库，则必须提供加密密钥，以便能够以任何方式使用数据库或事务日志。用于加密数据库的算法是由 -ea 选项指定的算法。如果您指定了 -ek 选项而没有指定 -ea，则使用 AES 算法。请参见“高度加密”一节第 1019 页。</p> <p>请保护密钥。请务必将密钥副本存储在一个安全的位置。丢失密钥将导致数据库完全无法访问，而且也无法对数据库进行恢复。</p>
<p>-ep</p>	<p>如果使用 -an 选项卸载并重装数据库，则可以使用此选项提示为新建的数据库指定加密密钥。这样，加密密钥决不会以明文显示，从而提供了额外的安全保证。如果指定 -ep 选项但不指定 -an 选项，则忽略 -ep 选项。如果指定 -ep 和 -an，则必须输入两次加密密钥，以确保正确地输入了密钥。如果两次输入的密钥不匹配，则卸载将失败。请参见“高度加密”一节第 1019 页。</p>

选项	说明
-er	<p>在卸载过程中，从加密表中删除加密。</p> <p>在重建已启用表加密的数据库时，必须指定 -er 或 -et 来指明新数据库是否启用表加密，否则在尝试向该新数据库装载数据时可能会出错。</p> <p>以下命令将拥有加密表的数据库 (<i>mydata.db</i>) 卸载到不启用表加密的新数据库 (<i>mydatacopy.db</i>) 中，即删除了所有加密表的加密：</p> <pre>dbunload -an c:\mydatacopy.db -er -c "UID=DBA;PWD=sql; DBF=c:\mydata.db; DBKEY=29bN8cjlz"</pre>
-et	<p>在新数据库中启用数据库表加密（还必须指定 -an 或 -ar）。如果指定 -et 选项但未指定 -ea 选项，将会使用 AES 算法。如果指定 -et 选项，就必须也指定 -ep 或 -ek。可以将新数据库的表加密设置更改为与所卸载的数据库不同的设置。</p> <p>在重建已启用表加密的数据库时，必须指定 -er 或 -et 来指明新数据库是否启用表加密，否则在尝试向该新数据库装载数据时可能会出错。</p> <p>下面的示例将含有简单加密算法加密表的数据库 (<i>mydata.db</i>) 卸载到启用表加密的新数据库 (<i>mydatacopy.db</i>) 中，采用 AES_FIPS 加密算法，密钥为 34jh：</p> <pre>dbunload -an c:\mydatacopy.db -et -ea AES_FIPS -ek 34jh -c "UID=DBA;PWD=sql;DBF=c:\mydata.db"</pre>
-g	<p>实例化视图 缺省情况下，执行重装操作后不会初始化定义为 MANUAL REFRESH 的实例化视图。如果想将这些实例化视图作为重装过程的一部分进行初始化，则应指定 -g 选项。指定 -g 选项会导致数据库服务器执行 <code>sa_refresh_materialized_views</code> 系统过程。请参见“sa_refresh_materialized_views 系统过程”一节《SQL Anywhere 服务器 - SQL 参考》。</p> <p>在决定是否使用 -g 选项时，需考虑到如果初始化所有实例化视图可能会大大延长完成重装过程的时间。另一方面，如果不使用 -g 选项，意味着第一个尝试使用未初始化的实例化视图的查询必须等待数据库服务器对该视图进行初始化，这可能会导致意外的延迟。如果不使用 -g 选项，也可以在重装完成后手工初始化实例化视图。请参见“初始化实例化视图”一节《SQL Anywhere 服务器 - SQL 的用法》。</p> <p>文本索引 缺省情况下，执行重装操作后不会初始化定义为 MANUAL REFRESH 的文本索引。如果想将这些文本索引作为重装过程的一部分进行初始化，则应指定 -g 选项。指定 -g 选项会导致数据库服务器执行 <code>sa_refresh_text_indexes</code> 系统过程。请参见“sa_refresh_text_indexes 系统过程”一节《SQL Anywhere 服务器 - SQL 参考》。</p>

选项	说明
-ii	使用 UNLOAD 语句从数据库中抽取数据，并使用 <i>reload.sql</i> 文件中的 LOAD 语句用数据重新填充数据库。这是缺省设置。
-ix	使用 UNLOAD 语句从数据库中抽取数据，并使用 <i>reload.sql</i> 文件中的 Interactive SQL INPUT 语句用数据重新填充数据库。
-k	<p>填充 <i>sa_diagnostic_auxiliary_catalog</i> 表。该表将表、用户、过程等数据库对象的 ID 从源数据库映射到跟踪数据库。它还会使所有直方图被卸载/重装。创建跟踪数据库（即，接收诊断跟踪信息的数据库）时，应使用此选项。<i>sa_diagnostic_auxiliary_catalog</i> 表使服务器可以模拟在捕获到跟踪数据时的各种条件（例如，用于索引顾问或应用程序分析）。与 -n 选项一起使用时，此选项非常有用。请参见“使用诊断跟踪进行高级应用程序分析”一节《SQL Anywhere 服务器 - SQL 的用法》和“<i>sa_diagnostic_auxiliary_catalog</i> 表”一节《SQL Anywhere 服务器 - SQL 参考》。</p>
-l	<p>在数据库重建期间，强制保留 SYSTABCOL.max_identity 的当前值。</p> <p>缺省情况下，在重新构建其中包含带有自动增量列的表的数据库时，数据库服务器会根据表的当前内容计算每个自动增量列的下一可用值。在大多数情况下，这已足够；但如果从值范围的末尾删除了若干行，则这些值可以重新使用，但在某些情况下这不是我们所希望的。</p> <p>指定 -l 选项会针对包含自动增量值的每个表，将对 sa_reset_identity 系统过程的调用添加到生成的 <i>reload.sql</i> 脚本中，从而保留 SYSTABCOL.max_identity 的当前值。</p> <p>另请参见：</p> <ul style="list-style-type: none"> ● “重装带有自动增量列的表”一节《SQL Anywhere 11 - 更改和升级》 ● “SYSTABCOL 系统视图”一节《SQL Anywhere 服务器 - SQL 参考》 ● “sa_reset_identity 系统过程”一节《SQL Anywhere 服务器 - SQL 参考》
-m	不保留在复制中所涉及的数据库的用户 ID。
-n	不卸载数据库中的数据； <i>reload.sql</i> 只包含用于构建数据库结构的 SQL 语句。如果希望 <i>reload.sql</i> 文件包含 LOAD TABLE 或 INPUT 语句，则应使用 -nl。
-nl	<p>卸载结构（与 -n 选项的行为相同），但结果 <i>reload.sql</i> 文件中还包含每个表的 LOAD TABLE 或 INPUT 语句。使用此选项时不会卸载任何用户数据。如果指定 -nl，还必须包括一个数据目录，以便可以生成 LOAD/INPUT 语句，即使没有文件写入该目录中。此选项使您可以在不卸载数据的情况下生成重装脚本。可以通过指定 -d 来卸载数据。如果数据库中的某个表包含不应卸载的数据，可以使用 dbunload -d -e table-name 跳过卸载该表的数据。</p>

选项	说明
-no	<p>卸载按名称排序的数据库对象。缺省情况下，dbunload 会按对象创建时的顺序来生成这些对象。在各数据库中包含相同的对象但这些对象的创建顺序不同时，指定 -no 选项会在比较数据库模式时很有用。</p> <p>如果指定 -no 选项，在 reload.sql 文件中会将对象定义按对象类型分组并按字母顺序排列：</p> <ul style="list-style-type: none"> ● 用户 ● 组成员 ● 表 ● 索引和外键 ● 视图 ● 过程 ● 函数 ● 触发器 ● 事件 ● Web 服务 <p>对象定义按照所有者、名称的顺序输出。有时候，在排序中还包含第三个元素（例如，外键角色名、触发器名）。</p> <p>-no 选项不能和 -n、-nl、-ar、-an 或 -ac 选项共用。为简化比较，建议在比较使用同一数据库服务器版本创建的数据库的重装脚本时，使用 -no 选项，因为二者的对象定义差异很小。</p> <div style="border: 1px solid black; padding: 5px; margin-top: 10px;"> <p>小心</p> <p>请勿使用生成的文件创建新数据库，因为在某些情况下对象的创建顺序很重要。例如，如果过程 p2 调用过程 p1 并且 p1 返回结果集，则在定义 p2 之前先定义 p1 就会很重要。试图执行使用 -no 选项生成的 reload.sql 脚本会导致发生错误。</p> </div>
-o filename	将输出消息写入指定的文件中。该文件的位置相对于 dbunload。
-p char	将外部卸载 (dbunload -x 选项) 的缺省转义字符 (\) 替换为其它字符。只有在命令提示符下运行此实用程序时才可以使用此选项。
-q	以安静模式运行—不显示消息或窗口。只有在命令提示符下运行此实用程序时才可以使用此选项。如果指定 -q，就必须也指定 -y。否则如果 reload.sql 已经存在，卸载就将失败。
-qc	指定在卸载完成时立即关闭消息窗口。缺省情况下，在用户执行关闭操作之前，dbunload 消息窗口会始终保持打开状态。此选项只能在 Windows Mobile 上使用。

选项	说明
-r <i>reload-file</i>	修改所生成重装命令文件的名称和目录。缺省值为当前目录中的 <i>reload.sql</i> 。此目录相对于客户端应用程序的当前目录，而不是相对于服务器。
-t <i>table, ...</i>	<p>指定要卸载的表的列表。缺省情况下，卸载所有表。此选项与 -n 选项一起使用时，将只卸载一组表定义。表名始终不区分大小写，即使在区分大小写的数据库中也如此。</p> <p>使用 -t 选项创建的 <i>reload.sql</i> 文件不应用于数据库的重建，因为该文件并不包含所有的数据库表。如果有外键引用某个表，则必须有该表的内容才能重建数据库。</p> <p>建议只将 -t 选项和 -d 选项一起使用，用以卸载除 -t 所指定表的数据。</p>
-u	如果您要卸载索引已损坏的数据库，使用此选项可避免使用已损坏的索引对数据进行排序。通常，如果表中定义了主键或聚簇索引，则每个表中的数据都会按主键或聚簇索引来排序。
-v	显示所卸载的表的名称以及已卸载的行数。只有在命令提示符下运行 dbunload 时才可以此选项。
-xi	通过将数据卸载到 dbunload 客户端，执行外部卸载，然后使用所生成重装命令文件 <i>reload.sql</i> 中的 LOAD 语句来将数据重新填充到数据库中。
-xx	通过将数据卸载到 dbunload 客户端，执行外部卸载，然后使用所生成重装命令文件 <i>reload.sql</i> 中的 Interactive SQL INPUT 语句来将数据重新填充到数据库中。
-y	<p>在不提示进行确认的情况下替换现有的命令文件。如果指定 -q，就必须也指定 -y。否则如果 dbunload 检测到命令文件已经存在，卸载就将失败。</p> <p>在卸载参与复制的数据库时，有一些特殊的注意事项。请参见“抽取远程数据库”一节《SQL Remote》和“升级 SQL Remote”一节《SQL Anywhere 11 - 更改和升级》。</p>
<i>directory</i>	指定用于放置卸载数据的目录。 <i>reload.sql</i> 命令文件始终相对于用户的当前目录。

注释

升级到版本 11

有关将现有数据库重建为版本 11 数据库的信息，请参见“升级 SQL Anywhere”一节《SQL Anywhere 11 - 更改和升级》。

在使用 **dbunload** 卸载版本 10.0.0 或更高版本的数据库时，所使用的 **dbunload** 的版本必须与用于访问数据库的数据库服务器的版本相匹配。如果将较旧版本的 **dbunload** 和相对较新版本的数据库服务器一起使用，会报告出现错误，反之亦然。

通过卸载实用程序，您可以卸载数据库，并将一组数据文件放在指定的目录中。卸载实用程序会创建一个 **Interactive SQL** 命令文件来重建您的数据库。它还将卸载每个表中的所有数据，并以逗号分隔格式将这些数据放到指定目录下的文件中。二进制数据通过转义序列来准确表示。

通过执行 **UPDATE ISYSUSER** 语句，内部卸载/重装操作会卸载有关每位用户当前状态的信息。外部卸载/重装不包含此类信息，并且会重置所有用户的状态。请参见“[管理登录策略概述](#)”一节第 416 页。

当通过卸载和重装来重建数据库时，重建的数据库可能会比原数据库小。数据库大小的减少可能是由于 **SQL Anywhere** 中的索引变化所致，并不表示出现了问题或有数据丢失。

注意

需要恢复的版本 9 及更早版本的数据库无法使用版本 10 或更高版本的卸载实用程序 (**dbunload**) 进行重新装载。必须使用版本 9 或更早版本的 **dbunload** 重新装载该数据库。

使用卸载实用程序也可以直接从现有数据库创建新的数据库。这样可以避免将数据库内容写入普通磁盘文件所带来的潜在安全问题。

如果您只想卸载表数据，则可以在 **Sybase Central** 中使用 **[卸载数据]** 窗口一步完成。

有关详细信息，请参见“[使用 \[卸载数据\] 窗口导出数据](#)”一节《[SQL Anywhere 服务器 - SQL 的用法](#)》。

在卸载参与复制的数据库时，有一些特殊的注意事项。请参见“[抽取远程数据库](#)”一节《[SQL Remote](#)》。

您可以以下列方式访问卸载实用程序：

- 从 **Sybase Central** 使用 **[卸载数据库向导]**。请参见“[使用 \[卸载数据库向导\] 导出数据](#)”一节《[SQL Anywhere 服务器 - SQL 的用法](#)》。
- 在命令提示符处，使用 **dbunload** 命令。如果要合并到批处理文件或命令文件中，此命令比较有用。

*卸载实用程序应由具有 **DBA** 权限的用户 ID 运行。只有这样，才能确保有权卸载所有数据。此外，**reload.sql** 文件也应由具有 **DBA** 权限的用户运行。（通常，它运行于只拥有用户 ID **DBA** 和口令 **sql** 的新数据库上。）*

数据库服务器 **-gl** 选项用于控制从数据库卸载数据所需要的权限。请参见“[-gl 服务器选项](#)”一节第 181 页。

dbo 用户 ID 在数据库中拥有一组系统对象，其中包括视图和存储过程。

卸载实用程序不会卸载在数据库创建期间为 **dbo** 用户 ID 所创建的对象。对这些对象进行的更改（例如，重新定义系统过程）会在数据库卸载后丢失。而在数据库初始化后由 **dbo** 用户 ID 所创建的所有对象都会被卸载实用程序卸载，因此这些对象会保留下来。

在卸载数据库时，任何对系统对象权限的更改都不会卸载。必须在新数据库中授予或撤消这些权限。

提示

在重建数据库之前，建议通过运行与以下命令类似的命令来重装不带任何数据的数据库以校验重装过程：

```
dbunload -n -an new.db -c "UID=your-user-id;PWD=your-password;DBF=original-database-file"
```

在重建数据库之前，必须修复在原数据库中发现的所有问题。

在缺省模式下，或在使用 `-ii` 或 `-ix` 时，`dbunload` 保存数据所使用的目录相对于数据库服务器，而不是相对于用户的当前目录。

如果使用 `-xi` 或 `-xx`，则此目录相对于用户的当前目录。

有关在此模式下提供文件名和路径的详细信息，请参见“UNLOAD 语句”一节《SQL Anywhere 服务器 - SQL 参考》。

如果不提供表的列表，则卸载整个数据库。如果提供表的列表，则只卸载这些表。

卸载的数据包括 `reload.sql` 文件中所生成的 LOAD TABLE 语句的列列表。卸载列列表便于对表中的列进行重新排序。各表可以删除或重新创建，然后使用 `reload.sql` 重新填充。

由 `dbunload` 生成的 LOAD TABLE 语句会关闭检查约束和计算列。

退出代码是零（成功）或非零（失败）。请参见“软件组件的退出代码”一节《SQL Anywhere 服务器 - 编程》。

具有实例化视图的数据库

建议在重建数据库后，在数据库中刷新实例化视图。请参见“刷新手动视图”一节《SQL Anywhere 服务器 - SQL 的用法》。

运行诊断跟踪的数据库

跟踪信息不会作为数据库卸载或重装操作的一部分被卸载。如果需要从一个数据库向另一个数据库传输跟踪信息，必须通过复制 `sa_diagnostic_*` 表的内容来手动执行；但是，不建议采用此方法。

内部与外部卸载和重装

以下选项提供了内外部卸载和重装的组合：`-ii`、`-ix`、`-xi` 和 `-xx`。相对于外部命令（Interactive SQL 的 INPUT 和 OUTPUT 语句），使用内部命令（UNLOAD/LOAD）会获得更高的性能。但是，内部命令由服务器执行，因此文件和目录路径相对于数据库服务器的位置。而使用外部命令时，文件和目录路径相对于用户的当前目录。

在 Sybase Central 中，您可以指定是相对于服务器还是相对于客户端进行卸载。请参见“UNLOAD 语句”一节《SQL Anywhere 服务器 - SQL 参考》。

当使用外部卸载和重装来卸载、重装或重建数据库时，如果此数据库的字符集与运行 `dbunload` 的主机系统的字符集不兼容，在此数据库的字符集与主机系统的字符集进行数据转换时，字符集转换可能会引起数据损坏。

为避免此问题，可在数据库的连接字符串中指定数据库的字符集（`-c` 和 `-ac` 选项）。例如，如果数据库字符集为 UTF-8，您应该在连接字符串中包括 `"charset=utf-8"`：

```
dbunload -c UID=user-ID;PWD=password;  
CHARSET=utf-8;DBF=filename -ac UID=user-ID;  
PWD=password;CHARSET=utf-8;ENG=server-name -xx
```

执行外部卸载时，*reload.sql* 的开头部分包括一条注释过的 CREATE DATABASE 语句。使用该语句，可以创建一个与正要卸载的数据库相当的数据库。

如果被卸载的数据库是使用 SQL Anywhere 版本 9 或更早的版本创建的，并且包含自定义归类，则 COLLATION 子句以如下形式出现：

```
COLLATION collation-label DEFINITION collation-definition
```

其中 *collation-definition* 是用于指定自定义归类的字符串。

保留自定义归类的唯一方法是以单步方式（内部卸载）重建数据库。如果选择卸载数据库，然后将模式和数据装载到您创建的数据库中，则必须使用所提供归类中的一种。

如果被卸载的数据库是通过高度加密的方法创建的，则 CREATE DATABASE 语句中的 KEY 子句的值会显示为三个问号 (???)。

失败的卸载

如果在使用 -ar 或 -an 进行数据库内部重建期间出现故障，则重装表数据并重建表上的所有索引后，dbunload 会在当前目录中创建一个名为 *unprocessed.sql* 的文件。该文件中包含所有因故障而未执行的语句，还包含引发故障的语句，以此作为一个注释。下面是 *unprocessed.sql* 文件的示例：

```
-- The database reload failed with the following error:  
-- ***** SQL error: the-SQL-ERROR  
-- This script contains the statements that were not executed as a  
-- result of the failure. The statement that caused the failure is  
-- commented out below. To complete the reload, correct the failing  
-- statement, remove the surrounding comments and execute this script.  
/*  
the failing statement  
go  
  
*/  
  
setuser "DBA"  
go  
  
... the remainder of the statements to be processed
```

有了此文件，您将有机会修正、删除或变更失败的语句。*unprocessed.sql* 文件仅在所有表数据和参照完整性约束重装后创建。使用 Interactive SQL，您可以连接到新数据库并执行更新后的 *unprocessed.sql* 文件。这样，您就无需再次启动重建即可完成数据库的重建，从而节约大量的时间。

生成 *unprocessed.sql* 文件后，dbunload 将停止并返回故障错误代码，以将这一失败的重建告知其它工具和脚本。

加密的数据库

在重建已启用表加密的数据库时，必须指定 -er 或 -et 来指明新数据库是否启用表加密，否则在尝试向该新数据库装载数据时可能会出错。

如果要卸载高度加密的数据库，则必须提供加密密钥。您可以使用 DatabaseKey (DBKEY) 连接参数在命令行上提供密钥。或者，如果您希望得到输入加密密钥的提示，而不是在普通视图中的命令行上输入密钥，则可以使用 `-ep` 服务器选项，如下所示：

```
dbunload -c "DBF=enc.db;START=dbeng11 -ep"
```

如果要使用 `-an` 选项来卸载数据库并重新装入新数据库中，并且要使用 `-ek` 或 `-ep` 选项来为新数据库设置加密密钥，请记住以下注意事项：

- 如果原数据库采用了高度加密，则需要使用 `-c` 选项中的 DatabaseKey (DBKEY) 连接参数来指定原数据库的密钥，而不是使用 `-ek` 或 `-ep` 选项。
- 使用 `-ek` 和 `-ep` 选项，可以卸载未加密的数据库并重装到高度加密的新数据库中。在使用 `-ep` 和 `-an` 时，必须正确地确认密钥，否则卸载就会失败。
- 如果原数据库是高度加密的，但是没有使用 `-ek` 和 `-ep` 选项，则可使用简单加密对新数据库进行加密。
- 如果不指定 `-an`，则忽略 `-ek` 和 `-ep` 选项。`dbunload -ek` 和 `-ep` 选项应用于新数据库，而数据库服务器 (`dbeng11/dbsrv11`) 选项和 `DBKEY=` 应用于现有数据库。
- 在重建同步或复制中所涉及的数据库时，`dbunload` 会假定使用 `-ek` 或 `-ep` 选项所指定的加密密钥是原始数据库及新建的数据库的加密密钥。

有关加密的详细信息，请参见“[-ep 服务器选项](#)”一节第 173 页和“[DatabaseKey 连接参数 \[DBKEY\]](#)”一节第 259 页。

重建数据库

要卸载数据库，首先要确保该数据库未在运行。然后，运行 `dbunload`，指定 DBA 用户和口令，通过 `DBF=` 连接参数引用该数据库。

要重装数据库，应创建一个新的数据库，然后通过 Interactive SQL 运行所生成的 `reload.sql` 命令文件。

要将卸载和重装步骤结合起来，除遵循上述有关卸载的说明外，还应添加 `-an` 选项以指定新数据库文件的名称。请参见有关 `-ac` 和 `-an` 选项的说明。

升级实用程序 (dbupgrad)

升级实用程序不能将数据库升级到版本 11

升级实用程序 (dbupgrad) 不能用于将版本 9.0.2 及更早版本的数据库升级到版本 11。要将版本 9.0.2 或更早版本的数据库升级到版本 11，必须通过执行卸载和重装来重建数据库。请参见“升级 SQL Anywhere”一节《SQL Anywhere 11 - 更改和升级》。

更新系统表和视图、添加新数据库选项及重新创建所有系统存储过程。在数据库中为 Java 安装 jConnect 支持和更改支持。

如果使用升级实用程序升级一个当前正在镜像的数据库服务器，将会返回一条错误消息。

语法

dbupgrad [options]

选项	说明
@data	从指定的环境变量或配置文件中读入选项。请参见“使用配置文件”一节第 695 页。 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“文件隐藏实用程序 (dbfhide)”一节第 724 页。
-c "keyword=value; ..."	指定连接参数。请参见“连接参数”一节第 248 页。 用户 ID 必须具有 DBA 权限。 例如，以下命令会升级称为 sample11 的数据库但不安装 jConnect 支持，它以用户 DBA 和口令 sql 进行连接： <pre>dbupgrad -c "UID=DBA;PWD=sql;DBF=c:\sa11\sample11.db" -i</pre>
-i	排除 jConnect 系统对象。如果要使用 jConnect JDBC 驱动程序来访问系统目录信息，则需要安装 jConnect 支持。指定此选项后，只要不访问系统信息，您就仍然可以使用 JDBC。以后需要 jConnect 支持时，您可以使用 Sybase Central 或 ALTER DATABASE UPGRADE 语句来添加它。请参见“在数据库中安装 jConnect 系统对象”一节《SQL Anywhere 服务器 - 编程》和“ALTER DATABASE 语句”一节《SQL Anywhere 服务器 - SQL 参考》。
-o filename	将输出消息写入指定文件中。
-q	以安静模式运行—不显示消息或窗口。

注释**小心**

在升级前，应始终对该数据库文件进行备份。如果在升级现有文件时升级失败，则这些文件将无法使用。有关备份数据库的信息，请参见“[备份和数据恢复](#)”第 817 页。

dbupgrad 实用程序可以升级使用早期版本软件创建的数据库，以便充分利用新版本的各项功能。其可以升级的最新版本是 SQL Anywhere 10.0.0。尽管较新版本的数据库服务器可以针对早期版本所创建的数据库运行，但如果不将数据库升级，在其创建后软件中所引入的某些功能就无法使用。

具有实例化视图的数据库

建议在升级数据库后，在数据库中刷新实例化视图。请参见“[刷新手动视图](#)”一节《[SQL Anywhere 服务器 - SQL 的用法](#)》。

可使用升级实用程序更新系统表和视图、添加新数据库选项、恢复数据库选项，并重新创建所有系统存储过程，以及在数据库中为 Java 安装 jConnect 支持和更改支持。

当针对 SQL Anywhere 的新版本和软件更新发布时，您可以通过升级实用程序来享用这些新的功能。

升级数据库不需要卸载再重装这些数据库。

如果您希望在升级的数据库上使用复制，则还必须将事务日志存档并在升级的数据库上开始使用新的事务日志。

您可以以下列方式访问升级实用程序：

- 从 Sybase Central 使用 [\[升级数据库向导\]](#)。
- 从 Interactive SQL 使用 ALTER DATABASE UPGRADE 语句。请参见“[ALTER DATABASE 语句](#)”一节《[SQL Anywhere 服务器 - SQL 参考](#)》。
- 在命令提示符处，使用 dbupgrad 命令。

退出代码是零（成功）或非零（失败）。请参见“[软件组件的退出代码](#)”一节《[SQL Anywhere 服务器 - 编程](#)》。

并不是所有的功能都变为可用

dbupgrad 并不能使需要对数据库文件进行物理重组的功能在升级后变得可用。这些功能包括索引增强和数据存储方面的改进。要想利用这些功能增强带来的好处，您必须卸载并重装数据库。请参见“[升级 SQL Anywhere](#)”一节《[SQL Anywhere 11 - 更改和升级](#)》。

校验实用程序 (dbvalid)

校验数据库中某些或所有表和实例化视图的索引和键。

语法

dbvalid [*options*] [*object-name*, ...]

选项	说明
@data	<p>从指定的环境变量或配置文件中读入选项。请参见“使用配置文件”一节第 695 页。</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“文件隐藏实用程序 (dbfhide)”一节第 724 页。</p>
-c "keyword=value; ..."	<p>指定数据库连接参数。有关连接参数的说明，请参见“连接参数”一节第 248 页。用户 ID 必须有 DBA 或 VALIDATE 权限。</p> <p>例如，以下命令会校验数据库 <i>c:\salesdata.db</i>，包括所有的表和实例化视图，它以用户 DBA 和口令 sql 进行连接：</p> <pre>dbvalid -c "UID=DBA;PWD=sql;DBF=c:\salesdata.db"</pre>
-d	<p>校验数据库中的所有表页是否属于正确的对象，并执行校验和校验。-d 选项不包括数据或索引的校验。-d 选项不能与 -i、-s 或 -t 选项一起使用。</p>
-fx	<p>校验表的每一行，并确保表中的行数与该表所关联的每个索引中的行数相匹配。此选项不为每个行执行单独的索引查找。在校验具有较小高速缓存的大数据库时，使用此选项可以大大提高性能。</p>
-i	<p>校验指定的索引。</p>
-o filename	<p>将输出消息写入指定的文件中。</p>
-q	<p>不将输出消息显示给客户端。但您仍可以使用 -o 选项将消息记录到文件中。</p>
-s	<p>使用校验和校验数据库。校验和用于确定是否在磁盘上已修改了数据库页。校验和校验读取磁盘上的每个数据库页，并计算其校验和。如果计算出来的校验和与页面存储的校验和不同，则该页面已被修改，并返回错误。任何无效页的页码都会显示在数据库服务器消息窗口中。-s 选项不能与 -d、-i、-t 或任何 -f 选项一起使用。</p>
-t	<p>指定 <i>object-name</i> 值的列表，该列表是表和实例化视图的列表。</p>
object-name	<p>指定要校验的表或实例化视图的名称。</p> <p>如果使用 -i，则 <i>object-name</i> 是指要校验的索引。</p>

注释

缺省情况下，dbvalid 会校验数据库中的所有表、实例化视图和索引，并且校验数据库本身。

通过校验实用程序，可以校验数据库中某些或所有表和实例化视图的索引和键。校验实用程序还可以用于验证数据库中的所有表页是否属于正确的对象，并验证页的校验和是否正确。缺省情况下，dbvalid 会校验数据库中的所有表和实例化视图（其行为与 -t 选项相同）。

对于每张表或每个实例化视图，校验实用程序都会扫描整个对象，然后按表上所定义的每个索引和键查找每个记录。校验实用程序还可以用于验证数据库中的所有表页是否属于正确的对象，并验证页的校验和是否正确。要运行校验实用程序，必须具有 DBA 或 VALIDATE 权限。

您还可以用下列方式访问校验实用程序：

- 从 Sybase Central 使用 [\[校验数据库向导\]](#)。请参见 [“校验数据库”](#) 一节第 859 页。
- 从 Interactive SQL 使用 VALIDATE 语句。请参见 [“VALIDATE 语句”](#) 一节 [《SQL Anywhere 服务器 - SQL 参考》](#)。

校验实用程序可与常规备份结合使用，以确保数据库中数据的完整性。如果要校验数据库的备份副本，建议先制作该备份的副本，然后校验副本。这样做可以确保不会更改在恢复期间使用的文件。请参见 [“备份和数据恢复”](#) 第 817 页。

小心

不得以任何方式更改数据库和事务日志的备份副本。如果在备份过程中未执行任何事务，或者指定了 BACKUP DATABASE WITH CHECKPOINT LOG RECOVER 或 WITH CHECKPOINT LOG NO COPY，则可以使用只读模式或通过验证备份数据库的副本来验证备份数据库的有效性。

但是，如果有正在执行的事务，或者指定了 BACKUP DATABASE WITH CHECKPOINT LOG COPY，则当您启动数据库服务器时，数据库服务器必须执行数据库恢复。恢复操作会修改备份副本，这会防止应用原数据库的后续事务日志文件。

如果运行校验实用程序时自动启动了某个数据库，则该数据库会以只读方式启动。在校验是备份或恢复计划的一部分时，这可防止对该数据库进行更改。

如果校验实用程序连接到某个正在运行的数据库，而该数据库不是以只读方式启动的，则校验实用程序会显示警告。该警告提醒您，不能将正在被校验的数据库用作恢复计划的一部分。执行备份的方式决定了通过 dbbackup 创建的大多数数据库都被标记为需要恢复。如果您正要校验的数据库需要恢复，且您希望强制该数据库以可读写方式启动，则可以在运行 dbvalid 之前启动该数据库，也可以为 DBS 连接参数指定一个有效值。请参见 [“DatabaseSwitches 连接参数 \[DBS\]”](#) 一节第 261 页。

如果 *mycopy.db* 数据库需要恢复，以下两个命令都允许运行 dbvalid:

```
dbvalid -c "UID=DBA;PWD=sql;DBF=mycopy.db;DBS=-n mycopy"
```

```
dbvalid -c "UID=DBA;PWD=sql;DBF=mycopy.db;DBS=-dh"
```

小心

应在没有任何连接对数据库进行更改时对表或整个数据库进行校验；否则，可能会报告错误，指出某种形式的数据库损坏，而实际上并没有任何损坏。

对于未启用校验和的数据库，校验实用程序会返回有关校验和冲突的警告。因为无论是否启用校验和，数据库服务器都会自动计算重要数据库页面的校验和。针对在 Windows Mobile 上运行的数据库或者在可能更不可靠的存储介质（例如可移动驱动器）上运行的数据库，数据库服务器也会自动为这些数据库创建校验和。请参见“[使用校验和检测损坏](#)”一节第 859 页。

校验要求以排它方式访问每个表。因此，最好在数据库中没有其它活动时进行校验。

退出代码是零（成功）或非零（失败）。请参见“[软件组件的退出代码](#)”一节《[SQL Anywhere 服务器 - 编程](#)》。

有关在校验期间进行的特定检查的详细信息，请参见“[VALIDATE 语句](#)”一节《[SQL Anywhere 服务器 - SQL 参考](#)》。

版本诊断实用程序 (dbversion)

返回有关指定的可执行文件的信息。

语法

dbversion *executable-name*

注释

实用程序仅能运行于 Unix 环境中，并返回 SQL Anywhere 可执行文件的信息。

另请参见

- “-v 服务器选项” 一节第 220 页

示例

以下命令：

```
$ dbversion /opt/sqlanywhere11/bin32/dbversion
```

返回 dbversion 可执行文件的信息：

```
SQL Anywhere Version Diagnostic Utility Version 11.0.1.1283
/opt/sqlanywhere11/bin32/dbversion: dbversion xx 11 0 1 1283 linux 2008/04/02
23:31:54
nothr 32 production
```

字段	说明
dbversion	返回可执行文件名。
xx	返回表示安装类型的双字母代码。
11	返回主版本号。
0	返回子版本号。
1	返回补丁号。
1180	返回内部版本号。
linux	返回操作系统代码。
2008/04/02 23:31:54	返回内部版本时间/日期戳。
nothr	返回线程模型（nothr 或 posix）。
32	返回可执行文件的位（32 或 64）。
production	返回生产或调试。

维护数据库

本节介绍如何备份数据库文件以及如何使用事件和调度来自动化数据库管理工作。

备份和数据恢复	817
校验数据库	857
使用调度和事件自动完成任务	863
SQL Anywhere 高可用性	879

备份和数据恢复

目录

备份快速入门	818
备份类型	819
选择备份格式	823
备份和恢复限制	824
创建服务器端备份	825
创建客户端备份	830
校验备份	832
恢复数据库	833
设计备份和恢复计划	843
备份参与同步和复制的数据库	847
内部备份过程	852

备份 是数据库中信息的完整或部分副本，保存在一个物理上独立的位置。当数据库变得不可用时，可以从备份中将其**恢复**。可以使用备份来将所有已提交更改恢复到数据库，直到数据库不可用。

如果对正在运行的数据库进行备份，则即使其他用户正在修改该数据库，也可以提供数据处于一致状态的数据库的快照。

如果操作系统或数据库服务器出现故障，或数据库服务器无法正常关闭，则必须恢复数据库。数据库服务器在数据库启动时会检查数据库是否在上一个会话结束时完全关闭。如果未关闭，则数据库服务器会执行自动恢复程序来恢复所有更改，直到最新提交的事务为止。

SQL Anywhere 工具创建对正在运行的数据库执行的**联机**备份。您必须具备 BACKUP 权限或 REMOTE DBA 权限才能进行数据库的联机备份。如果数据库未在运行，则可以通过复制数据库文件来创建**脱机**备份。

另请参见

- “备份类型”一节第 819 页
- “备份快速入门”一节第 818 页
- “备份实用程序 (dbbackup)”一节第 698 页
- “BACKUP 语句”一节 《SQL Anywhere 服务器 - SQL 参考》
- “设计备份和恢复计划”一节第 843 页
- “恢复数据库”一节第 833 页

备份快速入门

创建备份时，必须确定要存储备份文件的位置：数据库服务器计算机或客户端计算机。必须拥有 BACKUP 权限或 REMOTE DBA 权限才能通过以下步骤备份数据库。

◆ 创建服务器端备份

- 运行 BACKUP DATABASE 语句。例如：

```
BACKUP DATABASE DIRECTORY 'd:\\temp\\backup';
```

此语句在服务器计算机上的目录 `d:\temp\backup` 中创建数据库文件的备份副本。

或者，您可以运行带 `-s` 选项的 `dbbackup` 来创建备份。例如：

```
dbbackup -s -c "ENG=sample_server;DBN=demo;UID=DBA;PWD=sql"  
"c:\SQLAnybackup"
```

◆ 创建客户端备份

- 在客户端计算机上运行备份实用程序 (`dbbackup`)。例如：

```
dbbackup -c "ENG=sample_server;DBN=demo;UID=DBA;PWD=sql" "c:\SQLAnybackup"
```

另请参见

- “备份实用程序 (`dbbackup`)” 一节第 698 页
- “BACKUP 语句” 一节 《SQL Anywhere 服务器 - SQL 参考》
- “备份类型” 一节第 819 页
- “设计备份和恢复计划” 一节第 843 页
- “恢复数据库” 一节第 833 页

备份类型

下表总结了 SQL Anywhere 所支持的备份类型：

备份类型	说明	详细信息
联机	数据库运行期间所执行的备份。	请参见“ 联机和脱机备份 ”一节第 819 页。
脱机	数据库未运行期间所执行的备份。只有在数据库服务器已正确关闭后，才应执行此类备份。	请参见“ 联机和脱机备份 ”一节第 819 页。
完全	完全备份对数据库文件和事务日志都进行备份。通常，会在完全备份过程中穿插进行多个增量备份。	“ 完全备份 ”一节第 820 页
增量	仅事务日志备份。	“ 增量备份 ”一节第 821 页
活动	数据库运行期间对数据库进行的连续备份。	“ 活动备份 ”一节第 821 页
档案	包含备份所需全部信息的一个或多个文件的集合，其中包括主数据库文件、事务日志和任何其它 <code>dbspace</code> 。	“ 档案备份 ”一节第 823 页
映像	独立文件形式的数据库文件副本和/或事务日志副本。	“ 映像备份 ”一节第 823 页
服务器端	在数据库服务器计算机上创建的备份。	“ 创建服务器端备份 ”一节第 825 页
客户端	在客户端计算机上创建的备份。	“ 创建客户端备份 ”一节第 830 页

联机 and 脱机备份

脱机备份是数据库文件的副本。只有当数据库未运行且数据库服务器已正确关闭时，才应执行脱机备份。

SQL Anywhere 随附的所有工具（例如备份数据库实用程序 (dbbackup)、BACKUP DATABASE 语句和 Sybase Central 向导）都在数据库运行期间执行联机备份。

如果对正在运行的数据库进行备份，则即使其他用户正在修改该数据库，也可以提供数据处于一致状态的数据库的快照。

另请参见

- “[备份类型](#)”一节第 819 页

完全备份

完全备份对数据库文件和事务日志都进行备份。必须拥有 BACKUP 或 REMOTE DBA 权限才能执行完全备份。

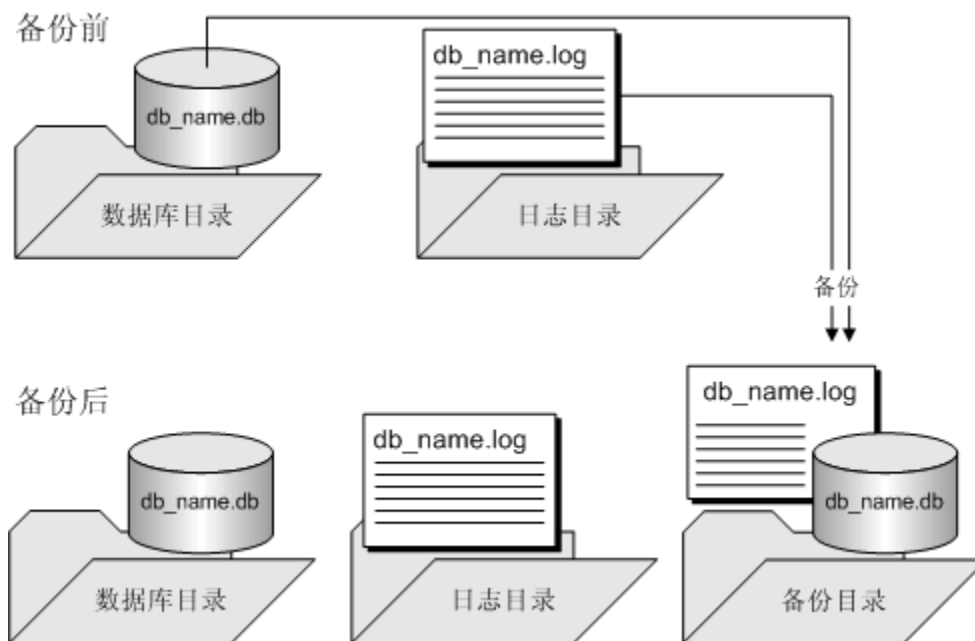
◆ 执行完全备份（概述）：

1. 对数据库执行校验检查以确保它未损坏。您可以使用校验实用程序或 sa_validate 存储过程。请参见“[校验数据库](#)”一节第 859 页。
2. 备份数据库文件和事务日志。

有关如何执行备份操作的信息，请参见：

- “[创建服务器端备份](#)”一节第 825 页
- “[创建客户端备份](#)”一节第 830 页
- “[执行备份和删除原始事务日志](#)”一节第 850 页
- “[执行备份和重命名原始事务日志](#)”一节第 848 页

最简单的备份方式是创建数据库文件和事务日志的副本，并使事务日志处于适当位置（未将其截断或替换）的映像备份（由独立文件形式的数据库文件和/或事务日志的副本组成）。任何备份都不改变数据库文件。下图说明了这种完全备份。



另请参见

- “[增量备份](#)”一节第 821 页
- “[备份类型](#)”一节第 819 页

增量备份

增量备份仅对事务日志进行备份。通常，会在完全备份过程中穿插进行多个增量备份。请参见“[完全备份](#)”一节第 820 页。

数据库文件和事务日志文件的备份副本与这些文件的联机版本同名。例如，如果您对示例数据库进行备份，则备份副本名为 *demo.db* 和 *demo.log*。当您重复备份语句时，选择一个新的备份目录以避免覆盖备份副本。

有关通过重命名事务日志的备份本来执行可重复增量备份命令的详细信息，请参见“[在备份过程中重命名事务日志的备份副本](#)”一节第 849 页。

◆ 执行增量备份（概述）：

1. 确保您对数据库拥有 BACKUP 或 REMOTE DBA 权限。
2. 备份事务日志，而不备份数据库文件。

另请参见

- “[创建服务器端备份](#)”一节第 825 页
- “[创建客户端备份](#)”一节第 830 页
- “[执行备份和删除原始事务日志](#)”一节第 850 页
- “[执行备份和重命名原始事务日志](#)”一节第 848 页
- “[完全备份](#)”一节第 820 页
- “[备份类型](#)”一节第 819 页

活动备份

活动备份是连续的数据库备份，有助于防止整机发生故障。可以使用事务日志的冗余副本来重新启动您辅助计算机上的系统。

如果系统出现故障，则可以使用备份的事务日志来快速重新启动系统。但是，根据数据库服务器正在处理的负载，活动备份可能会滞后，并且可能不会包含所有已提交的事务。

除活动备份之外还可以使用数据库镜像。请参见“[数据库镜像简介](#)”一节第 880 页。

通常，您应当从辅助计算机运行 `dbbackup` 实用程序。

如果主计算机变得不可用，则可以使用辅助计算机重新启动数据库。数据库文件和事务日志保存重新启动数据库所需的信息。

活动备份和定期备份

与活动事务日志相比，事务日志的活动备份总是与其等长或比其更短。当活动备份正在运行时，另一个备份重新启动事务日志（`dbbackup -r` 或 `dbbackup -x`），活动备份自动截断活动备份日志，并在开始新事务日志时重新启动活动备份。

另请参见

- “创建活动备份”一节第 830 页
- “从活动备份重新启动”一节第 836 页
- “备份类型”一节第 819 页

活动备份和事务日志镜像之间的区别

活动备份和事务日志镜像都提供事务日志的辅助副本。但是，在使用活动备份和使用事务日志镜像之间有几处不同：

- **通常，活动备份是针对不同的计算机创建的** 通过在单独计算机上运行备份实用程序，数据库服务器不对备份日志文件执行写操作，而数据传输由 SQL Anywhere 客户端/服务器通信系统来完成。因此，性能影响会减小，而可靠性会提高。
建议不在单独一台计算机上运行事务日志镜像。它可能导致性能和数据损坏问题，并在计算机之间的连接失败时停止数据库服务器。
- **活动备份针对不可用的计算机提供防护功能** 即使事务日志镜像保存在一个单独设备上，它也不会整机不可用时立即恢复。您可以考虑安排两台计算机共享对一组磁盘的访问权限。
- **活动备份可能滞后于数据库服务器** 事务日志镜像包含完全恢复已提交事务所需的所有信息。根据数据库服务器正在处理的负载，活动备份可能会滞后于事务日志镜像，并且不一定包含所有已提交事务。

选择备份格式

档案备份将数据库文件和事务日志复制到一个或多个文件中（通常复制到磁带驱动器上）。**映像备份**分别创建的数据库文件副本和/或事务日志副本。只能以服务器端备份的形式执行档案备份。

如果要直接备份到磁带上，则应使用档案备份。否则，应使用映像备份，因为映像备份更易于恢复。

档案备份

档案备份是包含备份所需全部信息的一个或多个文件的集合，其中包括主数据库文件、事务日志和任何其它 `dbspace`。只能以服务器端备份的形式执行档案备份。可将档案备份保存到文件或磁带驱动器。可以使用 `BACKUP DATABASE` 语句或 Sybase Central 中的 [\[备份数据库向导\]](#) 创建档案备份。

创建档案备份时，将向在 `BACKUP` 语句中为所创建的每一个文件指定的文件名中添加扩展名（.1、.2、.3 等）。

可以使用 Sybase Central 中的 [\[恢复数据库向导\]](#) 或使用 `RESTORE DATABASE` 语句从档案备份中恢复数据库。

档案备份仅在 Windows 和 Unix 平台上受到支持。在 Windows Mobile 上，只允许进行映像备份。请参见“[映像备份](#)”一节第 823 页。

有关创建档案备份的信息，请参见：

- “使用 [\[备份数据库向导\]](#)”一节第 827 页
- “使用 `BACKUP DATABASE` 语句创建服务器端备份”一节第 825 页
- “备份类型”一节第 819 页

映像备份

映像备份由独立文件形式的数据库文件副本和/或事务日志副本组成。

可使用备份实用程序 (`dbbackup`)、[\[创建备份映像向导\]](#) 或 `BACKUP DATABASE` 语句创建映像备份。可在所有受支持的平台上执行映像备份，它是 Windows Mobile 唯一支持的备份类型。

如果要在磁带中创建备份，则使用档案备份。请参见“[档案备份](#)”一节第 823 页。

另请参见

- “`BACKUP` 语句”一节 [《SQL Anywhere 服务器 - SQL 参考》](#)
- “备份实用程序 (`dbbackup`)”一节第 698 页
- “`RESTORE DATABASE` 语句”一节 [《SQL Anywhere 服务器 - SQL 参考》](#)
- “使用 `BACKUP DATABASE` 语句创建服务器端备份”一节第 825 页
- “使用 [\[创建备份映像向导\]](#)”一节第 828 页
- “备份类型”一节第 819 页

备份和恢复限制

数据库服务器禁止在备份过程中执行以下操作：

- 另一个备份（活动备份除外）。
- 检查点（不包括由备份指令发出的检查点）。
- 任何导致出现检查点的语句。这包括数据定义语句、LOAD TABLE 语句和 TRUNCATE TABLE 语句。

在恢复过程（包括恢复备份）中，不允许数据库的其他用户执行任何操作。

另请参见

- [“了解检查点日志”一节第 17 页](#)

创建服务器端备份

在数据库服务器计算机上创建备份通常比在客户端计算机上快，因为数据不必通过客户端/服务器通信系统进行传输。要在您的应用程序中生成服务器端备份，请使用 SQL 语句。以下方法可用于创建服务器端备份：

工具	详细信息
BACKUP 语句	“使用 BACKUP DATABASE 语句创建服务器端备份 ”一节第 825 页
备份实用程序 (dbbackup)	“使用备份实用程序 (dbbackup) 创建服务器端备份”一节第 827 页
备份数据库向导	“使用 Sybase Central 创建服务器端备份 ”一节第 827 页
创建备份映像向导	“使用 Sybase Central 创建服务器端备份 ”一节第 827 页
创建维护计划向导	“使用 Sybase Central 创建服务器端备份 ”一节第 827 页
DBBackup 函数	“ a_backup_db 结构 ”一节 《 SQL Anywhere 服务器 - 编程 》
SQL Anywhere 卷影复制服务 (dbvss)	“使用 SQL Anywhere 卷影复制服务 (VSS) ”一节第 829 页

备份实用程序 (dbbackup) 和 BACKUP DATABASE 语句都使用物理设备级并行操作来减少完成备份操作所需的时间。Windows Mobile 不支持并行备份。请参见“[了解并行数据库备份](#)”一节第 855 页。

另请参见

- “[创建客户端备份](#)”一节第 830 页
- “[备份类型](#)”一节第 819 页

使用 BACKUP DATABASE 语句创建服务器端备份

本主题介绍保持事务日志不变的备份。有关创建备份时其它事务日志管理选项的信息，请参见“[BACKUP 语句](#)”一节 《[SQL Anywhere 服务器 - SQL 参考](#)》。

BACKUP 语句在与数据库服务器可执行文件位于同一个目录中的 *backup.syb* 文本文件中创建一个条目。

创建映像备份

◆ 创建映像备份 (SQL)

- 使用以下子句来执行备份语句:

```
BACKUP DATABASE  
DIRECTORY directory-name;
```

有关从映像备份进行恢复的信息, 请参见 [“从映像备份恢复”](#) 一节第 834 页。

创建档案备份

◆ 创建到磁带的档案备份 (SQL):

- 使用包含以下子句的 BACKUP 语句:

```
BACKUP DATABASE  
TO archive-root  
[ ATTENDED { ON | OFF } ]  
[ WITH COMMENT comment-string ];
```

如果您将 ATTENDED 选项设置为 OFF, 则在磁带或磁盘空间不足时, 备份会失败。如果将 ATTENDED 设置为 ON, 则会在备份档案设备没有更多的空间时提示您采取相应措施。

有关从档案备份进行恢复的信息, 请参见 [“从档案备份恢复”](#) 一节第 835 页。

示例

以下语句为当前数据库和事务日志创建映像备份, 并将其保存到不同的文件中, 然后重命名现有事务日志。

```
BACKUP DATABASE  
DIRECTORY 'd:\\temp\\backup'  
TRANSACTION LOG RENAME;
```

以下语句在 Windows 计算机的第一个磁带驱动器中创建档案备份:

```
BACKUP DATABASE  
TO '\\\\.\\tape0'  
ATTENDED OFF  
WITH COMMENT 'May 6 backup';
```

Windows 上的第一个磁带驱动器是 \\.\tape0。因为反斜线在 SQL 字符串中为转义字符, 所以每个反斜线前面还有一个反斜线。

另请参见

- [“BACKUP 语句”](#) 一节 《SQL Anywhere 服务器 - SQL 参考》
- [“档案备份”](#) 一节第 823 页
- [“映像备份”](#) 一节第 823 页
- [“恢复数据库”](#) 一节第 833 页

使用备份实用程序 (dbbackup) 创建服务器端备份

本主题介绍保持事务日志不变的备份。有关创建备份时其它事务日志管理选项的信息，请参见“[备份实用程序 \(dbbackup\)](#)”一节第 698 页。

dbbackup 实用程序创建由独立文件形式的数据库文件和/或事务日志副本组成的映像备份。

◆ 执行备份，继续使用原来的事务日志（命令行）

- 如果要使用 dbbackup 实用程序，请使用以下语法：

```
dbbackup -c "connection-string" [ -t ] backup-directory
```

只有当执行增量备份时才使用 -t 选项。请参见“[增量备份](#)”一节第 821 页。

示例

以下示例在数据库服务器计算机的目录 `c:\SQLAnybackup` 中创建备份。

```
dbbackup -s -c "ENG=sample_server;DBN=demo;UID=DBA;PWD=sql" "c:\SQLAnybackup"
```

另请参见

- “[映像备份](#)”一节第 823 页
- “[从映像备份恢复](#)”一节第 834 页
- “[恢复数据库](#)”一节第 833 页

使用 Sybase Central 创建服务器端备份

要从 Sybase Central 创建服务器端备份，请使用以下向导之一：

- **备份数据库向导** 此向导创建档案备份。可以指定用于存储该备份的文件名或磁带驱动器。请参见“[使用 \[备份数据库向导\]](#)”一节第 827 页。
- **创建备份映像向导** 此向导在数据库运行时创建各个数据库文件的副本。要进行恢复，可以将所有文件重新复制到其在数据库服务器计算机上的原始位置。请参见“[使用 \[创建备份映像向导\]](#)”一节第 828 页。
- **创建维护计划向导** 此向导允许您为多个任务（包括备份数据库）创建调度。您可以选择创建档案备份、文件映像备份或增量备份。请参见“[创建维护计划](#)”一节第 845 页。

另请参见

- “[档案备份](#)”一节第 823 页
- “[映像备份](#)”一节第 823 页
- “[增量备份](#)”一节第 821 页

使用 [备份数据库向导]

[备份数据库向导] 创建档案备份。当您在 Sybase Central 中创建档案备份时，您可以选择将数据库直接备份到磁带或是磁盘上。

◆ 从 Sybase Central 创建备份 ([备份数据库向导])

1. 以具有 BACKUP 或 REMOTE DBA 权限的用户身份连接到数据库。
2. 右击数据库并选择 [备份数据库]。
3. 请按照向导中的说明进行操作。

另请参见

- “档案备份”一节第 823 页
- “从档案备份恢复”一节第 835 页
- “恢复数据库”一节第 833 页

使用 [创建备份映像向导]

[创建备份映像向导] 为各个数据库文件创建副本。要进行恢复，可以将所有文件重新复制到其在数据库服务器计算机上的原始位置。

该过程介绍最简单的备份类型，它未触及事务日志。

◆ 执行备份，继续使用原来的事务日志 (Sybase Central):

1. 以具有 BACKUP 或 REMOTE DBA 权限的用户身份连接到数据库。
2. 右击数据库并选择 [创建备份映像]。
3. 单击 [下一步]。
4. 在 [您要备份哪个数据库] 列表中，选择数据库并单击 [下一步]。
5. 在 [将备份映像保存在以下目录中] 字段中，键入目录名称以保存备份副本。
6. 在 [您要备份哪些文件] 列表中选择选项并单击 [下一步]。
7. 在 [您需要对事务日志进行什么操作] 列表中，单击 [继续使用同一事务日志]。
8. 单击 [下一步]。
9. 单击 [完成]。
10. 单击 [关闭]。

提示

还可以使用以下方法从 Sybase Central 访问 [创建备份映像数据库向导]:

- 选择一个数据库，然后选择 [文件] » [创建备份映像]。
- 选择 [工具] » [SQL Anywhere 11] » [创建备份映像]。

另请参见

- “映像备份”一节第 823 页
- “从映像备份恢复”一节第 834 页
- “恢复数据库”一节第 833 页

使用 SQL Anywhere 卷影复制服务 (VSS)

SQL Anywhere 与 Microsoft 卷影复制服务 (Volume Shadow Copy Service, 简称 VSS) 兼容。VSS 可用于创建整个磁盘卷或卷集的时间点快照和制作文件副本, 这些文件是要独占打开以供诸如 SQL Anywhere 数据库服务器等应用程序独占使用的文件。32 位 Windows XP 操作系统、32 位和 64 位的 Windows 2003 及更高版本的操作系统 (包括 Windows Vista) 都支持 VSS。

缺省情况下, 如果 SQL Anywhere VSS 写入程序 (*dbvss11.exe*) 正在运行, 则所有 SQL Anywhere 数据库都能使用 VSS 服务进行备份。您可以使用没有 SQL Anywhere VSS 写入程序的 VSS 对数据库进行备份。但是, 您可能需要使用完整的 SQL Anywhere 恢复过程来恢复这些数据库。要使数据库服务器不参与 VSS 服务, 请在启动数据库服务器时包括 **-vss-**。或者, 可以使用 Windows 服务实用程序 (*dbsvc*) 来指定何时启动 VSS 服务。

VSS 如何与 SQL Anywhere 协同工作:

- 您的备份应用程序向 VSS 发送命令来拍摄快照。
- VSS 发出 **identify** 命令到 SQL Anywhere VSS 写入程序 (*dbvss11.exe*)。
- VSS 发出 **prepare to snapshot** 命令以挂起所有事务, 并将所有修改过的页面写入到所有数据库服务器上所有数据库的磁盘中。如果某数据库上的事务未在 10 秒内挂起, 则快照可能包含未提交的事务, 因此有必要进行完全恢复。
- VSS 发出 **freeze** 命令到检查点, 然后挂起所有数据库服务器上所有数据库的所有活动。每个 SQL Anywhere 数据库服务器等待所有数据库挂起所有活动的最长时间为 60 秒。通常, 此过程只需几秒钟。
- VSS 发出 **thaw** 命令到 SQL Anywhere VSS 写入程序, 以恢复所有数据库服务器上所有数据库的所有事务。

在极少数情况下, SQL Anywhere 可能无法在 VSS 允许的最长时间内挂起事务或完成检查点操作。如果出现这种情况, 您必须使用事务日志文件和完全恢复过程来恢复备份数据库。

另请参见

- “用于 Windows 的服务实用程序 (*dbsvc*)”一节第 773 页
- “创建 Windows 服务”一节第 60 页

创建客户端备份

可以使用备份实用程序 (dbbackup) 在客户端计算机上创建备份。备份实用程序 (dbbackup) 使用物理设备级并行操作来减少完成备份操作所需的全部时间。Windows Mobile 不支持并行备份。请参见“了解并行数据库备份”一节第 855 页。

以下方法可用于创建客户端备份：

工具	详细信息
备份实用程序 (dbbackup)	“使用 dbbackup 实用程序创建客户端备份”一节第 830 页
DBBackup 函数	“a_backup_db 结构”一节 《SQL Anywhere 服务器 - 编程》

使用 dbbackup 实用程序创建客户端备份

dbbackup 实用程序创建由独立文件形式的数据库文件和/或事务日志副本组成的映像备份。

◆ 创建客户端备份 (dbbackup)

- 在客户端计算机上运行备份实用程序 (dbbackup)。例如：

```
dbbackup -c "ENG=sample_server;DBN=demo;UID=DBA;PWD=sql" SQLAnybackup
```

另请参见

- “备份实用程序 (dbbackup)”一节第 698 页
- “创建服务器端备份”一节第 825 页
- “恢复数据库”一节第 833 页
- “备份类型”一节第 819 页

创建活动备份

可以使用活动备份来提供事务日志的冗余副本。当运行数据库服务器的主系统变得不可用时，可以使用此副本重新启动辅助系统。活动备份连续运行，只在服务器关闭时才终止。如果系统出现故障，则可以使用备份的事务日志来快速重新启动系统。但是，根据服务器正在处理的负载，活动备份可能会滞后，并且可能不会包含所有已提交的事务。

应从辅助计算机运行 dbbackup 实用程序。如果主计算机变得不可用，则可以使用辅助计算机重新启动数据库。数据库文件和事务日志保存重新启动所需的信息。

您可以使用带有 -l 选项的 dbbackup 实用程序对事务日志执行活动备份。

◆ 创建活动备份 (dbbackup 实用程序)

1. 设置辅助计算机，在联机计算机出现故障时，可从该计算机运行数据库。确保在辅助计算机上装有 SQL Anywhere。
2. 定期完全备份到辅助计算机。

例如：

```
dbbackup -c "UID=DBA;PWD=sql;ENG=testsrv;DBN=test;LINKS=tcPIP" c:\backup
```

3. 对事务日志运行活动备份，将其备份到辅助计算机。

```
dbbackup -l path\filename.log -c "connection-string"
```

4. 通常从辅助计算机运行 `dbbackup` 实用程序。

如果主计算机变得不可用，则可以使用辅助计算机重新启动数据库。数据库文件和事务日志保存重新启动所需的必需信息。

另请参见

- “活动备份” 一节第 821 页
- “恢复数据库” 一节第 833 页
- “从活动备份重新启动” 一节第 836 页
- “备份类型” 一节第 819 页

校验备份

在数据库服务器尝试访问数据库的受影响部分之前，数据库文件的损坏情况可能并未得到确认。作为您备份和恢复计划的一部分，应通过使用诸如 Sybase Central 中的 [\[校验数据库向导\]](#) 或校验实用程序 (dbvalid) 之类的工具定期检查数据库是否有效。在执行备份之前和之后均应校验数据库。必须拥有 VALIDATE 权限才能执行校验活动。请参见 [“VALIDATE 特权”一节第 427 页](#)。

当启动数据库的备份副本以对其进行校验时，可使用 -ds 数据库选项指定 dbspace 文件和事务日志的位置。这可允许您启动原数据库所在计算机上的数据库的备份副本，同时原数据库保持运行。请参见 [“-ds 数据库选项”一节第 237 页](#)。

根据指定的选项，校验可以包括校验和、索引数据的正确性，以及数据库中的所有表页是否属于对象。快速数据库校验 (-fx 选项) 不校验数据、连续行结构或外键关系。

小心

不得以任何方式更改数据库和事务日志的备份副本。如果在备份过程中未执行任何事务，或者指定了 BACKUP DATABASE WITH CHECKPOINT LOG RECOVER 或 WITH CHECKPOINT LOG NO COPY，则可以使用只读模式或通过验证备份数据库的副本来验证备份数据库的有效性。

但是，如果有正在执行的事务，或者指定了 BACKUP DATABASE WITH CHECKPOINT LOG COPY，则当您启动数据库服务器时，数据库服务器必须执行数据库恢复。恢复功能会修改备份副本，这是不希望出现的情况。

如果您能够确定在备份过程中未执行任何事务，则数据库服务器无需执行恢复步骤。在这种情况下，您可以使用只读数据库选项对备份执行有效性检查。请参见 [“-r 服务器选项”一节第 204 页](#)。

提示

使用 BACKUP 语句时结合 WAIT BEFORE START 子句可确保在开始备份时没有任何事务在执行。

校验要求以独占方式访问进行校验的对象。因此，最好在数据库中无其它活动时进行校验。

如果数据库文件中的基表损坏，则将其视为介质故障，并从以前的备份中恢复。如果索引损坏，最好先卸载没有索引的数据库，然后重装。

另请参见

- [“校验数据库”一节第 859 页](#)
- [“校验事务日志”一节第 851 页](#)
- [“校验表”一节第 860 页](#)
- [“VALIDATE 语句”一节 《SQL Anywhere 服务器 - SQL 参考》](#)
- [“在校验数据库时提高性能”一节第 862 页](#)
- [“恢复数据库”一节第 833 页](#)

恢复数据库

恢复是恢复数据库文件、事务日志和 dbspace，并通过增量事务日志文件使数据库文件尽可能处于最新状态的过程。

作为备份和恢复计划的一部分校验备份是很重要的。只应从数据库的有效备份副本恢复。

恢复过程中需要采取的步骤取决于在备份过程的增量备份中是否保持事务日志不变。如果您的备份操作删除或重命名事务日志，则可能要从多个事务日志中应用更改。如果备份操作未更改事务日志，则只需使用联机事务日志进行恢复。

如果有多个事务日志，则事务可能会跨越多个事务日志。恢复时**必须**按正确的顺序应用事务日志；否则跨多个事务日志的事务会被回退。如果希望由数据库服务器确定应用事务日志的正确顺序，则可以指定 `-ad` 数据库服务器选项。请参见“[通过多个事务日志恢复数据库](#)”一节第 837 页。

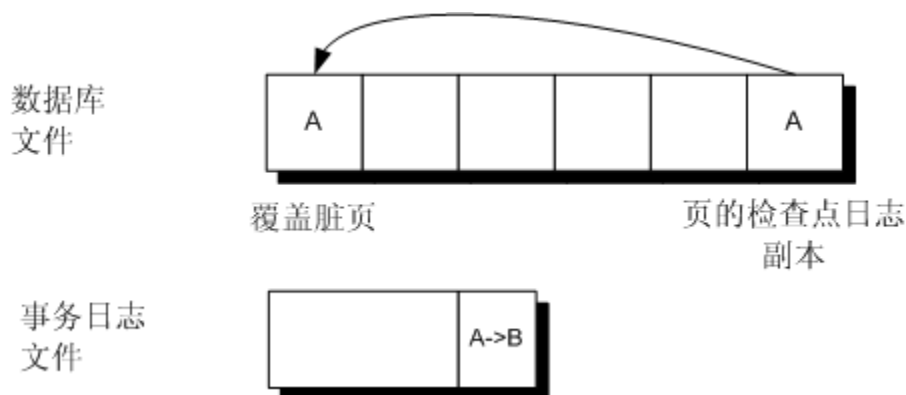
自动恢复过程

当数据库在正常操作过程中关闭时，数据库服务器会执行一个检查点操作，以使数据库中的所有信息都保存在数据库文件中。这就是**净关机**。

每次启动数据库时，数据库服务器都会检查上次关机是净关机还是由于系统出现故障所致。如果数据库并非净关机，它会自动执行以下步骤以便从系统故障中恢复：

1. 恢复到最近的检查点

要将所有页恢复到其在最近检查点处的状态，则复制检查点日志页来覆盖自该检查点以来进行的更改。



2. 应用自该检查点以来进行的更改

应用在该检查点和系统故障之间进行的更改（保存在事务日志中）。

3. 回退未提交的事务

使用回退日志回退任何未提交的事务。

恢复未提交的操作

当从数据库文件的介质故障中恢复时，事务日志保持不变。恢复过程将所有已提交的事务重新应用到数据库中。在某些情况下，您可能希望查找有关在发生故障时未完成的事务的信息。

[[翻译日志文件向导](#)] 会帮助您从 Sybase Central 将日志文件翻译为 `.sql` 文件。您还可以使用 `dbtran` 实用程序将日志文件翻译为 `.sql` 文件。

◆ 从事务日志中恢复未提交的操作 (Sybase Central):

1. 选择 [工具] » [SQL Anywhere 11] » [[翻译日志文件](#)]。
2. 请按照向导中的说明进行操作。
3. 在文本编辑器中编辑翻译的日志 (SQL 命令文件) 并标识所需的指令。

◆ 从事务日志中恢复未提交的操作 (命令行)

1. 运行 `dbtran`，将事务日志转换为 SQL 命令文件，并使用 `-a` 选项包括未提交的事务。例如，以下命令使用 `dbtran` 转换事务日志：

```
dbtran -a sample.log changes.sql
```

2. 在文本编辑器中编辑翻译的日志 (SQL 命令文件) 并标识所需的指令。

有关翻译日志文件实用程序的详细信息，请参见“[翻译日志文件实用程序 \(dbtran\)](#)”一节第 753 页。

注意

事务日志可能包含也可能不包含截至发生故障时的更改。不过它确实包含在最近提交的对数据库进行更改的事务结束之前进行的任何更改。

另请参见

- “[从映像备份恢复](#)”一节第 834 页
- “[从档案备份恢复](#)”一节第 835 页
- “[从活动备份重新启动](#)”一节第 836 页
- “[通过多个事务日志恢复数据库](#)”一节第 837 页
- “[从介质故障中恢复](#)”一节第 839 页

从映像备份恢复

以下过程假定您没有作为恢复过程的一部分进行应用所需的任何事务日志的增量备份。有关已备份多个事务日志副本后恢复数据库的信息，请参见“[通过多个事务日志恢复数据库](#)”一节第 837 页。

◆ 从映像备份恢复数据库

1. 将数据库文件重新复制到其原始位置。

- 重新启动数据库服务器。

另请参见

- “映像备份”一节第 823 页
- “自动恢复过程”一节第 833 页
- “恢复未提交的操作”一节第 834 页
- “备份类型”一节第 819 页

从档案备份恢复

以下过程假定您没有作为恢复过程的一部分进行应用所需的任何事务日志的增量备份。有关已备份多个事务日志副本后恢复数据库的信息，请参见“通过多个事务日志恢复数据库”一节第 837 页。

◆ 从档案备份中恢复数据库 (Sybase Central):

- 启动个人数据库服务器。

例如，以下命令用于启动名为 restore 的数据库服务器：

```
dbeng11 -n restore
```

- 启动 Sybase Central 并连接到实用程序数据库。
 - 在 [连接] 窗口的 [标识] 选项卡上输入用户 ID **DBA** 和口令 **sql**。将该选项卡上的所有其它字段留空。
 - 单击 [数据库] 选项卡，然后输入数据库名称 **utility_db**。将该选项卡上的所有其它字段留空。
 - 单击 [确定]。
- 选择 [工具] » [SQL Anywhere 11] » [恢复数据库]。
- 请按照向导中的说明进行操作。

◆ 从档案备份中恢复数据库 (Interactive SQL):

- 启动个人数据库服务器。

例如，以下命令用于启动名为 restore 的数据库服务器：

```
dbeng11 -n restore
```

- 启动 Interactive SQL 并连接到实用程序数据库。
 - 在 [连接] 窗口的 [标识] 选项卡上输入用户 ID **DBA** 和口令 **sql**。将该选项卡上的所有其它字段留空。
 - 单击 [数据库] 选项卡，然后输入数据库名称 **utility_db**。将该选项卡上的所有其它字段留空。
 - 单击 [确定]。
- 执行 RESTORE DATABASE 语句，同时指定档案根。

此时，您可以选择将档案数据库恢复到它的原始位置（缺省值），或使用 RENAME 子句恢复到具有不同设备名的不同计算机上。请参见“RESTORE DATABASE 语句”一节《SQL Anywhere 服务器 - SQL 参考》。

示例

以下语句将数据库从磁带档案恢复到数据库文件 `c:\newdb\newdb.db` 中。

```
RESTORE DATABASE 'c:\\newdb\\newdb.db'  
FROM '\\\\.\\tape0';
```

以下语句将数据库从文件 `c:\backup\archive.1` 中的档案备份恢复到数据库文件 `c:\newdb\newdb.db`。事务日志的名称和位置是在数据库中指定的。

```
RESTORE DATABASE 'c:\\newdb\\newdb.db'  
FROM 'c:\\backup\\archive';
```

另请参见

- “使用实用程序数据库”一节第 28 页
- “档案备份”一节第 823 页
- “自动恢复过程”一节第 833 页
- “恢复未提交的操作”一节第 834 页
- “备份类型”一节第 819 页

从活动备份重新启动

活动备份是在与运行生产数据库的主计算机之外的计算机上创建的备份。要从活动备份中重新启动数据库，您的辅助计算机上必须装有 SQL Anywhere。有关活动备份的详细信息，请参见“活动备份”一节第 821 页。

◆ 使用活动备份重新启动数据库：

1. 将完全备份事务日志文件和活动备份事务日志复制到一个目录之中，在该目录中这些日志文件可应用于数据库文件的备份副本。
2. 如果当前事务日志文件名称与预期的事务日志文件名称（如果有）相同，则重命名或删除当前事务日志文件。
3. 使用 `-ad` 选项启动数据库服务器，以应用步骤 1 所创建的目录中的事务日志并使数据库成为最新：

```
dbeng11 samples-dir\demo.db -ad directory-name
```

在应用了事务日志之后，数据库服务器自动关闭。

4. 按照正常方式启动数据库服务器，允许用户访问。任何新活动将写入到新的事务日志。
5. 对事务日志运行活动备份，将其备份到辅助计算机。

```
dbbackup -l path\filename.log -c "connection-string"
```

另请参见

- “活动备份”一节第 821 页
- “自动恢复过程”一节第 833 页
- “恢复未提交的操作”一节第 834 页
- “备份类型”一节第 819 页

通过多个事务日志恢复数据库

如果您需要恢复数据库并且有多个事务日志，则必须将事务日志文件按照正确的顺序应用到数据库的备份副本。

可使用以下任一方法来按正确顺序应用事务日志：

- 使用 `-a` 服务器选项将每个日志分别应用至数据库的备份副本。可以使用事务日志实用程序 (`dblog`) 确定生成事务日志文件的顺序。该实用程序生成并显示事务日志中的最早日志偏移，这可作为确定应用多个日志文件的顺序的有效方法。请参见“`-a` 数据库选项”一节第 234 页。
- 使用 `-ad` 服务器选项来指定事务日志文件的位置。数据库服务器根据日志偏移确定将事务日志应用至数据库的备份副本的正确顺序。请参见“`-ad` 数据库选项”一节第 234 页。
- 使用 `-ar` 服务器选项来使数据库服务器应用与事务日志所在目录中的数据库相关联的日志文件。事务日志位置从数据库中获得。数据库服务器根据日志偏移确定将事务日志应用至数据库的备份副本的正确顺序。请参见“`-ar` 数据库选项”一节第 235 页。
- 使用翻译日志文件实用程序 (`dbtran`) 将一个或多个事务日志翻译为可应用于数据库备份副本的 `.sql` 文件。请参见“事务日志实用程序 (`dblog`)”一节第 792 页。

使用 `-ad` 服务器选项来通过多个事务日志恢复数据库

可使用 `-ad` 服务器选项，通过将指定目录中的所有事务日志应用到数据库的备份副本来恢复数据库。指定该选项之后，数据库服务器会应用事务日志，然后关闭数据库。

◆ 使用 `-ad` 服务器选项从多个事务日志中恢复：

- 使用 `-ad` 启动数据库服务器，以便将事务日志应用于数据库的备份副本。请参见“`-ad` 数据库选项”一节第 234 页。

示例

以下示例使用 `-ad` 数据库服务器选项将脱机（备份）和当前事务日志应用到示例数据库的备份副本。数据库服务器使用事务日志中的日志偏移来确定应用日志文件的正确顺序。

1. 将备份事务日志和当前事务日志复制到一个目录之中，如 `c:\backuplogs`。
2. 启动数据库服务器并将事务日志应用到名为 `backupdemo.db` 的数据库的备份副本中：

```
dbeng11 backupdemo.db -ad c:\backuplogs
```

该数据库服务器将事务日志应用于数据库的备份副本，并随后关闭。

另请参见

- “使用 `-a` 服务器选项来通过多个事务日志恢复数据库” 一节第 838 页
- “使用 `dbtran` 实用程序来通过多个事务日志恢复数据库” 一节第 838 页
- “自动恢复过程” 一节第 833 页
- “恢复未提交的操作” 一节第 834 页

使用 `-a` 服务器选项来通过多个事务日志恢复数据库

使用 `-a` 服务器选项可以通过将单个事务日志文件应用于数据库的备份副本来恢复数据库。指定该选项之后，数据库服务器会应用日志，然后关闭。如果拥有多个事务日志，则必须按照正确的顺序（从最旧到最新）每次应用一个日志。

◆ 使用 `-a` 服务器选项从多个事务日志中恢复：

1. 使用 `-a` 启动数据库服务器，以便将备份事务日志应用于数据库的脱机（备份）副本。
请参见“`-a` 数据库选项”一节第 234 页。
2. 启动数据库服务器并将当前事务日志应用于数据库的备份副本。

示例

以下示例使用 `-a` 数据库服务器选项将脱机（备份）事务日志和当前事务日志应用于示例数据库的备份副本。

1. 启动数据库服务器并将名为 `backupdemo.log` 的备份事务日志应用于名为 `backupdemo.db` 的数据库的备份副本：

```
dbeng11 backupdemo.db -a backupdemo.log
```

该数据库服务器将备份事务日志应用于数据库的备份副本，并随后关闭。

2. 启动数据库服务器并将名为 `demo.log` 的当前事务日志应用于数据库的备份副本：

```
dbeng11 backupdemo.db -a demo.log
```

该数据库服务器将当前事务日志应用于数据库的备份副本，并随后关闭。

另请参见

- “使用 `-ad` 服务器选项来通过多个事务日志恢复数据库” 一节第 837 页
- “使用 `dbtran` 实用程序来通过多个事务日志恢复数据库” 一节第 838 页
- “自动恢复过程” 一节第 833 页
- “恢复未提交的操作” 一节第 834 页

使用 `dbtran` 实用程序来通过多个事务日志恢复数据库

为了在使用 `dbtran` 翻译多个事务日志时保持数据的完整性，必须同时指定 `-m` 和 `-n` 选项。`-m` 选项指示翻译日志文件实用程序 (`dblog`) 生成一个包含指定目录内的日志中的所有事务的文件（由 `-n` 命名）。

需要使用 `-m`，因为如果使用 `dbtran` 分别翻译每个日志，则跨越事务日志文件的任何事务都可能被回退。当 `dbtran` 翻译日志时，在日志的末尾处将添加 `ROLLBACK` 语句以撤消任何未提交的事务。针对事务跨越两个日志的情况，会在第二个日志文件中发生对事务的 `COMMIT`。第一个日志文件末尾处的操作将被 `dbtran` 回退，因为该文件不包含对事务的 `COMMIT`。使用 `-m` 翻译目录中的所有事务日志文件可以确保所有事务都会被翻译。请参见“[事务日志实用程序 \(dblog\)](#)”一节第 792 页。

◆ 使用 `dbtran` 实用程序从多个事务日志中恢复：

1. 针对包含事务日志文件的目录运行翻译日志文件实用程序 (`dbtran`)，并将生成的 SQL 语句输出到 `.sql` 文件中。
2. 启动数据库的备份副本。
3. 在 Interactive SQL 中，将步骤 1 中由 `dbtran` 生成的 `.sql` 文件应用于数据库的备份副本。

示例

以下示例使用 `dbtran` 实用程序将备份事务日志和当前事务日志应用于数据库的备份副本。

1. 针对 `c:\backup` 目录运行翻译日志文件实用程序，并将 SQL 语句输出到名为 `recoverylog.sql` 的文件中：

```
dbtran -m "c:\backup" -n recoverylog.sql
```

2. 启动名为 `backupdemo.db` 的数据库的备份副本：

```
dbeng11 backupdemo.db
```

3. 在 Interactive SQL 中，将 `recoverylog.sql` 文件应用于数据库：

```
dbisql -c "UID=DBA;PWD=sql;ENG=backupdemo" READ recoverylog.sql
```

另请参见

- “[使用 `-ad` 服务器选项来通过多个事务日志恢复数据库](#)”一节第 837 页
- “[使用 `-a` 服务器选项来通过多个事务日志恢复数据库](#)”一节第 838 页
- “[自动恢复过程](#)”一节第 833 页
- “[恢复未提交的操作](#)”一节第 834 页

从介质故障中恢复

如果数据库无法使用，则说明数据库出现了故障。SQL Anywhere 针对以下类型的故障提供保护：

故障类型	说明	示例
介质	数据库文件和/或事务日志变得无法使用。此类故障可能是由于文件系统或存储数据库文件的设备变得不可用或文件损坏而造成的。备份针对介质故障保护数据。	<ul style="list-style-type: none"> ● 存储数据库文件或事务日志文件的磁盘驱动器变得不可用。 ● 数据库文件或事务日志文件损坏。这可能是由于硬件问题或软件问题而造成的。
系统	<p>如果在计算机或操作系统出现故障时有部分完成的事务，则会出现系统故障。此类故障可能是由于计算机的关闭或重新启动不当、另一个应用程序导致操作系统出现故障或者电源故障而造成的。</p> <p>在系统出现故障之后，数据库服务器在您下次启动数据库时自动恢复。在出现系统错误之前提交的每个事务的结果都保持不变。在系统出现故障前未提交的事务进行的所有更改都被取消。</p>	<ul style="list-style-type: none"> ● 存在可能是由于电源故障、操作系统故障或计算机重新启动不当而导致的部分完成的事务时，计算机或操作系统变得暂时不可用。

从数据的介质故障中恢复

此过程介绍当您丢失的唯一文件是数据库时从介质故障进行恢复的步骤。

◆ 从数据库文件的介质故障中恢复：

1. 创建当前事务日志的额外备份副本。因为数据库文件不可用，所以事务日志包含自上次备份以来所做的更改的唯一记录。
2. 创建**恢复目录**以保存在恢复过程中使用的文件。
3. 将数据库文件从上次完全备份中复制到恢复目录中。
4. 将保留在备份事务日志中的事务应用于恢复数据库。使用以下方法之一。

要手工应用每个事务日志，请按时间顺序针对每个日志文件执行以下操作：

- a. 将日志文件复制到恢复目录中。
- b. 用应用事务日志 (-a) 选项启动数据库服务器以便应用事务日志：

```
dbeng11 database-name.db -a log-name.log
```

一旦应用事务，数据库服务器即自动关闭。

- c. 应用了所有备份的事务日志之后，将联机事务日志复制到恢复目录中。
将联机事务日志中的事务应用于恢复数据库。

```
dbeng11 database-name.db -a log-name.log
```

如果您希望由数据库服务器确定事务日志的正确顺序并自动应用它们，请执行以下操作：

- a. 将脱机和联机事务日志文件复制到恢复目录中。
- b. 用 `-ad` 选项启动数据库服务器，以指定事务日志的位置。数据库服务器根据日志偏移确定应用事务日志的正确顺序：

```
dbeng11 database-name.db -ad log-directory
```

一旦应用事务，数据库服务器即自动关闭。

5. 对恢复数据库执行有效性检查。
请参见“[校验数据库](#)”一节第 859 页。
6. 创建备份。
7. 将数据库文件移至生产目录中。
8. 通知用户，他们可以访问生产数据库。

另请参见

- [“自动恢复过程”一节第 833 页](#)
- [“恢复未提交的操作”一节第 834 页](#)

从事务日志镜像的介质故障中恢复

以下过程说明了使用事务日志镜像时如何从介质故障中恢复。如果您的数据库是复制服务器安装中的主站点，或者是 SQL Remote 安装中的统一数据库，则应使用事务日志镜像或等同的硬件。

◆ 从事务日志镜像的介质故障中恢复

1. 对在启动事务日志时提取的数据库文件创建额外的备份副本。
2. 确定这两个文件中的哪个受损。在事务日志及其镜像上运行翻译日志文件实用程序 (dbtran)。生成错误消息的文件损坏。翻译日志文件实用程序可从 Sybase Central 访问，或可作为 dbtran 实用程序访问。

以下命令行翻译名为 *demo.log* 的事务日志，并将转换后的输出放在 *demo.sql* 中：

```
dbtran demo.log
```

翻译日志文件实用程序正确翻译完整的文件，并在翻译受损文件时报告错误。

3. 复制正确的文件来覆盖损坏的文件。
4. 重新启动数据库服务器。

另请参见

- [“事务日志镜像”一节第 15 页](#)
- [“自动恢复过程”一节第 833 页](#)
- [“恢复未提交的操作”一节第 834 页](#)

从未镜像事务日志的介质故障中恢复

如果您的数据库是复制服务器安装中的主站点，或者是 MobiLink 或 SQL Remote 安装中的统一数据库，则应使用事务日志镜像或等同的硬件。请参见“[事务日志镜像](#)”一节第 15 页。

◆ 从未镜像事务日志的介质故障中恢复（部分恢复）：

1. 创建数据库文件的额外备份副本。缺少事务日志时，数据库文件包含自上次备份和最近的检查点之后所做的更改的唯一记录。
2. 删除或重命名事务日志文件。
3. 使用 -f 选项重新启动数据库。

```
dbeng11 samples-dir\demo.db -f
```

小心

只有当数据库不参与 MobiLink、SQL Remote 或复制服务器系统时，才应使用此命令。如果您的数据库是 SQL Remote 复制系统中的统一数据库，则可能必须重新抽取远程数据库。

如果没有 -f 选项，数据库服务器会将缺少事务日志报告为错误。利用 -f 选项，数据库服务器将数据库恢复到最近的检查点，然后回退在检查点时未提交的任何事务。新事务日志随之创建。

另请参见

- [“事务日志”一节第 14 页](#)
- [“自动恢复过程”一节第 833 页](#)
- [“恢复未提交的操作”一节第 834 页](#)
- [“-f 恢复选项”一节第 175 页](#)

设计备份和恢复计划

建议您制订并执行备份调度以保护您的数据。还应确保您已将备份和恢复命令作为备份和恢复计划的一部分进行了创建和测试。

制定备份和恢复计划时需要考虑的一些因素包括：

- 数据库文件位于什么位置？
- 需要备份什么文件？
- 备份文件存储在什么位置？
- 备份如何影响数据库或应用程序的性能？
- 当您进行备份时，数据库服务器将会正在运行吗？

需要进行备份的一些最常见情况包括：

- 介质故障
- 硬件故障
- 文件损坏

备份通常使用完全备份和增量备份的组合。每一种备份类型的频率取决于要保护的数据的类型。还应校验备份以确保其可用于恢复。请参见“[校验备份](#)”一节第 832 页。

可以使用 SQL Anywhere 中的调度功能来自动执行数据库备份任务。在您指定了调度之后，数据库服务器会自动执行备份。请参见“[使用调度和事件自动完成任务](#)”第 863 页和“[创建维护计划](#)”一节第 845 页。

贵组织在不访问数据库中数据的情况下运行的时间长度决定了最长恢复时间。

您应当确认能否在出现介质故障时为数据库文件和事务日志文件提供所需的保护。如果要在复制环境中运行，则应考虑使用事务日志镜像。请参见“[防止出现介质故障](#)”一节第 846 页。

外部因素可能会影响恢复时间，这些因素包括可用的硬件、数据库文件的大小、恢复介质、磁盘空间和意外错误等。计划备份策略时，应为诸如输入恢复命令或检索和装载磁带之类的任务留出额外的恢复时间。

另请参见

- [“了解备份”一节第 852 页](#)
- [“备份参与同步和复制的数据库”一节第 847 页](#)
- [“备份和恢复限制”一节第 824 页](#)
- [“备份类型”一节第 819 页](#)
- [“选择备份格式”一节第 823 页](#)
- [“完全备份”一节第 820 页](#)
- [“增量备份”一节第 821 页](#)

执行备份和恢复计划

◆ 执行备份和恢复计划：

1. 创建并校验备份和恢复命令，包括数据库校验命令。请参见 [“校验备份”一节第 832 页](#)。
2. 测量它执行备份和恢复命令所花费的时间。
3. 记录备份命令并创建书面的执行步骤，以描述备份的保存位置。这些过程应标识所使用的任何命名约定以及所执行的备份类型。
4. 在生产服务器上设置备份过程。
5. 监视备份过程以避免发生意外错误。确保备份过程中的任何变更均反映在文档中。

另请参见

- [“完全备份”一节第 820 页](#)
- [“增量备份”一节第 821 页](#)
- [“了解备份”一节第 852 页](#)
- [“备份参与同步和复制的数据库”一节第 847 页](#)
- [“备份和恢复限制”一节第 824 页](#)
- [“备份类型”一节第 819 页](#)
- [“选择备份格式”一节第 823 页](#)

调度注意事项

备份通常使用完全备份和增量备份的组合。创建备份的频率取决于数据的重要性和数据更改的频率等因素。

常见的备份出发点是每周执行完全备份，每天对事务日志进行增量备份。完全备份和增量备份都可以在服务器端或客户端联机（数据库正在运行时）或脱机执行。

备份调度提供保护所针对的故障种类不仅取决于备份调度的频率，而且还取决于数据库服务器的运行方式。

您始终都应保留多个完全备份。*如果创建备份时覆盖上次备份，在备份过程中万一出现介质故障，会使您根本留不下备份。*您还应当在现场之外保留某些完全备份，以防发生火灾、水灾、地震、偷盗或故意破坏。

您可以使用 SQL Anywhere 的事件调度功能，在调度的时间自动执行联机备份。请参见 [“创建维护计划”一节第 845 页](#)。

创建维护计划

要简化管理，可为您的数据库设置由数据库服务器自动执行的维护计划。维护计划由一个执行一项或多项下列任务的调度组成：

- 校验数据库
- 备份数据库
- 管理维护计划报告

在 Sybase Central 中使用 **[创建维护计划向导]** 创建维护计划。一次只能运行一个维护计划实例。每次运行维护计划时，数据库中都会保存一份维护计划报告。可以从 Sybase Central 中查看该报告，也可以选择在对数据库执行完维护计划之后，通过电子邮件将报告发送给您。

自定义维护计划

维护计划可以包含用户定义的操作。在 **[创建维护计划向导]** 中，可将用户定义的操作添加为在校验之前或备份之后运行的 SQL 语句。

创建维护计划报告

◆ 创建维护计划

1. 以具有 DBA 权限的用户身份连接到数据库。
2. 在左窗格中，右击 **[维护计划]** 并选择 **[新建]** » **[维护计划]**。
3. 请按照向导中的说明进行操作。

有关可用设置的信息，请参见：

- “定义调度”一节第 866 页
- “VALIDATE 语句”一节 《SQL Anywhere 服务器 - SQL 参考》
- “备份类型”一节第 819 页
- “xp_startsmtp 系统过程”一节 《SQL Anywhere 服务器 - SQL 参考》

另请参见

- “档案备份”一节第 823 页
- “映像备份”一节第 823 页
- “增量备份”一节第 821 页
- “创建维护计划”一节第 845 页

查看维护计划报告

维护计划执行完毕后，您可以在 Sybase Central 中查看其报告。

◆ 查看维护计划报告 (Sybase Central)

1. 通过 SQL Anywhere 插件以具有 DBA 权限的用户身份连接到数据库。

2. 在左窗格中，双击 [维护计划]。
3. 双击维护计划。
4. 在右窗格中，双击报告。

随即出现 [维护计划报告属性] 窗口。[详细信息] 窗格包含维护计划的日志。

防止出现介质故障

备份针对介质故障保护数据。

创建数据库时，事务日志的缺省位置是与数据库文件相同的设备和目录。这种安排不能针对介质故障提供保护，因此应考虑将事务日志放在其它位置以供生产使用。

数据库文件出现介质故障 如果您的数据库文件不可用，但事务日志可用，则只要您的备份过程正确，就可以恢复所有已提交的数据库更改。自数据库文件上次备份副本以来的所有信息都保存在备份事务日志或者联机事务日志中。

事务日志文件出现介质故障 除非您使用事务日志镜像，否则不能恢复在上次数据库检查点和事务日志出现介质故障这段时间间隔内输入的信息。因此，建议您在安装配置（如 SQL Remote 统一数据库）中使用事务日志镜像，因为在这种安装配置中，如果事务日志丢失，则可能会导致丢失关键信息或者中断复制系统。

从介质故障中恢复的速度取决于是数据库文件还是事务日志文件出现介质故障。

为了针对介质故障提供全面的保护，您应当将事务日志与数据库文件分别放在不同的设备上。某些有两个或多个硬盘驱动器的计算机只有一个物理磁盘驱动器，其中含有多个逻辑驱动器或分区：如果要为介质故障提供可靠保护，请确保您的计算机至少有两个物理存储设备。

如果将事务日志放在另一个设备上，则还可以通过避免在事务日志和主数据库文件之间移动磁盘磁头来改善性能。

小心

不应将事务日志放在网络目录上。通过网络读写页面会导致性能很差，并可能损坏文件。

另请参见

- “创建数据库”一节第 20 页
- “更改事务日志的位置”一节第 15 页

备份参与同步和复制的数据库

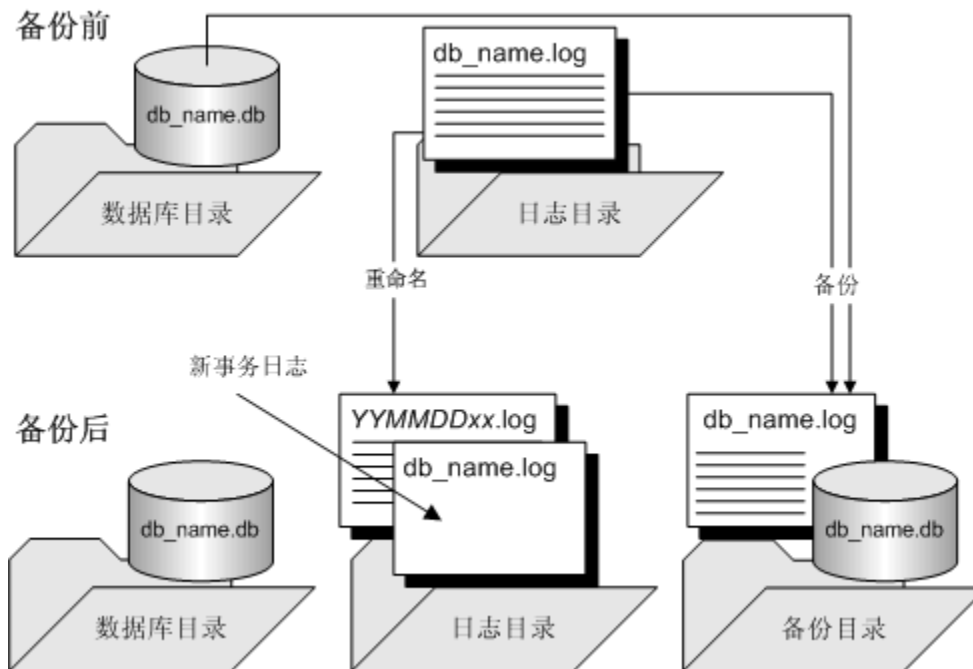
如果您的数据库属于 SQL Remote 安装的一部分，则消息代理必须能够访问旧事务。如果它是统一数据库，则它保留整个 SQL Remote 安装的主副本，并且必须执行整个备份过程，以确保数据不会丢失。

如果您的数据库是 [复制服务器] 安装的主站点，则 [复制代理] 需要能够访问旧事务。但是，磁盘空间限制常常使事务日志无限增大变得不切实际。

如果您的数据库参与使用 dbmlsync 进行的 MobiLink 安装，则需要注意相同的事项。但是，如果您的数据库是 MobiLink 统一数据库，则不需要旧事务日志。

在同步和复制环境中，您可以选择备份选项来重命名和重新启动事务日志。此类备份可防止事务日志无限增大，同时维护有关旧事务的信息。

下图阐释这种备份。



有关详细信息，请参见“[执行备份和重命名原始事务日志](#)”一节第 848 页。

备份过程在远程数据库上并不像在统一数据库上那样重要。您可以选择将复制统一数据库作为数据备份方法。一旦介质发生故障，必须从统一数据库中重新抽取远程数据库，任何未复制的操作都将丢失。（您可以使用翻译日志文件实用程序来尝试恢复丢失的操作。请参见“[翻译日志文件实用程序 \(dbtran\)](#)”一节第 753 页。

即使您确实选择依靠复制来保护远程数据库数据，也仍需定期在远程数据库上进行备份，以防止事务日志变得过大。您应该使用在统一数据库中使用的那些选项（重命名和重新启动日志）运行消息代理，使其能够访问已重命名的日志文件。如果您在远程数据库中将 `delete_old_logs` 选项设置为 On，则当不再需要旧事务日志时，消息代理会自动将其删除。

在 SQL Remote 中自动重命名事务日志

如果使用 -x 消息代理选项，则当数据库服务器关闭时，无需重命名远程计算机上的事务日志。-x 选项在事务日志被扫描以检查是否有外发消息之后将对该事务日志进行重命名。请参见“[消息代理 \(dbremote\)](#)”一节《SQL Remote》。

管理事务日志

备份数据库时，必须确定是要继续使用现有事务日志还是要创建新的事务日志。如果您的数据库参与了同步或复制，则必须保留旧事务日志的副本，直到您确定不再需要它们。

执行备份和重命名原始事务日志

◆ 执行备份，重命名事务日志 (Sybase Central):

1. 以具有 BACKUP 权限的用户身份连接到数据库。
2. 右击数据库并选择 [创建备份映像]。
3. 单击 [下一步]。
4. 在 [您要备份哪个数据库] 列表中，选择数据库并单击 [下一步]。
5. 在 [将备份映像保存在以下目录中] 字段中，键入目录名称以保存备份副本。
6. 在 [您要备份哪些文件] 列表中选择选项并单击 [下一步]。
7. 在 [您需要对事务日志进行什么操作] 列表中，单击 [重命名事务日志]。
8. 单击 [下一步]。
9. 单击 [完成]。
10. 单击 [关闭]。

◆ 执行备份，重命名事务日志 (SQL):

- 使用包含以下子句的 BACKUP 语句:

```
BACKUP DATABASE  
  DIRECTORY backup-directory  
  [ TRANSACTION LOG ONLY ]  
  TRANSACTION LOG RENAME;
```

只有当执行增量备份时才包括 TRANSACTION LOG ONLY 子句。

事务日志和数据库文件的备份副本位于 *backup-directory* 中。如果您输入一个路径，则它相对于数据库服务器（而非客户端应用程序）的工作目录。

◆ 执行备份，重命名事务日志（命令行）

- 运行以下命令（此命令必须在一行中键入）：

```
dbbackup -c "connection-string" -r [ -t ] backup-directory
```

如果执行增量备份，请包括 `-t` 选项。

事务日志和数据库文件的备份副本位于 *backup-directory* 中。如果您输入一个路径，则它相对于在其中运行该命令的目录。

另请参见

- “备份实用程序 (dbbackup)” 一节第 698 页
- “BACKUP 语句” 一节 《SQL Anywhere 服务器 - SQL 参考》
- “事务日志” 一节第 14 页
- “恢复数据库” 一节第 833 页

在备份过程中重命名事务日志的备份副本

在缺省情况下，事务日志文件的备份副本与联机文件同名。对于每个备份操作，您必须为备份副本指派一个不同的名称或位置，或者必须在完成下一个备份前移动备份副本。

要创建可重复的增量备份命令，可重命名事务日志的备份副本。

◆ 重命名事务日志的备份副本 (SQL):

- 在 BACKUP 语句中使用 MATCH 关键字。例如，以下语句在 *c:\backup* 目录中创建事务日志的增量备份。事务日志的备份副本名为 *YYMMDDxx.log*，其中 *YYMMDD* 是日期，*xx* 是从 AA 开始的计数器。

```
BACKUP DATABASE
DIRECTORY 'c:\\backup'
TRANSACTION LOG ONLY
TRANSACTION LOG RENAME MATCH;
```

◆ 重命名事务日志的备份副本 (命令行)

- 为 dbbackup 提供 `-n` 选项。例如，以下命令对示例数据库进行增量备份，并重命名事务日志的备份副本。

```
dbbackup -c "UID=DBA;PWD=sql;DBN=demo" -r -t -n c:\backup
```

注意

事务日志的备份副本名为 *YYMMDDxx.log*，其中 *YY* 是年份，*MM* 是月份，*DD* 是该月的第几天，*xx* 为 AA 到 ZZ（在每天有多个备份时递增）。*YYMMDDxx.log* 文件名可用于区分文件，而不用于排序。

这组备份选项通常用于包括在复制中的数据库。除了创建数据库文件和事务日志的备份副本以外，还将备份时的事务日志重命名为脱机日志，并启动一个与备份时使用的日志同名的新事务日志。

另请参见

- “备份参与同步和复制的数据库”一节第 847 页
- “备份实用程序 (dbbackup)”一节第 698 页
- “BACKUP 语句”一节《SQL Anywhere 服务器 - SQL 参考》
- “事务日志”一节第 14 页
- “恢复数据库”一节第 833 页

执行备份和删除原始事务日志

如果您的数据库未参与到复制中，并且您计算机上的磁盘空间有限，则可以在执行备份时删除联机事务日志的内容（**截断**日志）。要在使用此类备份时恢复您的数据库，则必须在从数据库文件的介质故障中恢复的过程中，使用自上次完全备份以来创建的每个备份副本。

◆ 执行备份，删除事务日志 (Sybase Central):

1. 以具有 DBA 权限的用户身份连接到数据库。
2. 右击数据库并选择 [创建备份映像]。
3. 单击 [下一步]。
4. 在 [您要备份哪个数据库] 列表中，选择数据库并单击 [下一步]。
5. 在 [将备份映像保存在以下目录中] 字段中，键入目录名称以保存备份副本。
6. 在 [您要备份哪些文件] 列表中选择选项并单击 [下一步]。
7. 在 [您需要对事务日志进行什么操作] 列表中，单击 [截断事务日志]。
8. 单击 [下一步]。
9. 单击 [完成]。
10. 单击 [关闭]。

◆ 执行备份，同时删除事务日志 (SQL):

- 使用包含以下子句的 BACKUP 语句:

```
BACKUP DATABASE  
  DIRECTORY backup-directory  
  [ TRANSACTION LOG ONLY ]  
  TRANSACTION LOG TRUNCATE;
```

只有当执行增量备份时才包括 TRANSACTION LOG ONLY 子句。

事务日志和数据库文件的备份副本位于 *backup-directory* 中。如果您输入一个路径，则它相对于数据库服务器（而非客户端应用程序）的工作目录。

◆ 执行备份，删除事务日志（命令行）

- 运行以下命令:

```
dbbackup -c "connection-string" -x [ -t ] backup-directory
```

只有当执行增量备份时才使用 `-t` 选项。

事务日志和数据库文件的备份副本位于 `backup-directory` 中。如果您输入一个路径，则它相对于在其中运行该命令的目录。

另请参见

- “备份实用程序 (dbbackup)” 一节第 698 页
- “BACKUP 语句” 一节 《SQL Anywhere 服务器 - SQL 参考》
- “事务日志” 一节第 14 页
- “恢复数据库” 一节第 833 页

校验事务日志

当使用事务日志镜像的数据库启动时，数据库服务器执行一系列检查和自动恢复操作，以便确认事务日志及其镜像未受损，并在检测到损坏纠正一些问题。

在启动时，服务器通过对事务日志及其镜像进行完全比较来检查这两个文件是否相同；如果它们相同，则数据库按通常的方式启动。比较日志和镜像会增加数据库的启动时间。

如果数据库由于系统故障而停止，则可能是因为在有些操作写入了事务日志而没有写入镜像。如果服务器发现：事务日志和镜像这两文件相比较，一直到其中较短的一个文件的结尾，这两个文件都是相同的，那么，较长文件的其余部分就会复制到较短的文件中。这将生成相同的日志和镜像。在自动执行该恢复步骤之后，服务器按通常的方式启动。

如果在检查过程中发现事务日志和事务日志镜像的正文部分不同，则说明其中某个文件受损。在这种情况下，数据库不启动，并生成一则错误消息，说明事务日志或其镜像无效。

无论具有联机事务日志还是脱机事务日志，都可以使用翻译日志文件实用程序 (dbtran) 校验事务日志。如果翻译日志文件实用程序可以成功地读取日志文件，则说明它是有效的。请参见“[翻译日志文件实用程序 \(dbtran\)](#)” 一节第 753 页。

内部备份过程

本节介绍备份过程中使用的内部机制。

执行备份时会发生什么情况？

当您启动备份时，数据库可能正由许多用户使用。如果要从备份恢复数据库，则需要知道哪些信息已备份，哪些信息尚未备份。

执行备份时，数据库服务器：

1. 发出检查点。备份完成之前不允许有更多检查点。
2. 执行完全备份时创建数据库文件的备份。
3. 创建事务日志的备份。

该备份包括在读取最后一页日志之前记录在事务日志中的所有操作。这可能包括备份启动之后发出的指令。

事务日志的备份副本通常小于联机事务日志。数据库服务器向联机事务日志分配的空间大小为 64 KB 的整数倍，因此事务日志文件的大小通常包括空页。但是，只备份非空页。

4. 标记数据库的备份映像，以表明需要进行恢复。当数据库备份副本已启动时，此步骤会导致自应用备份开始以来所发生的所有操作。如果尚未提交检查点的未完成操作，则会撤消这些操作。

了解备份

当数据库完全关闭时，数据库文件保存数据库中所有数据的最新完整副本。但是，当数据库正在运行时，数据库文件通常不是最新或完整的。

只有在刚完成检查点操作时，才能确保数据库文件保存所有数据的最新完整副本。在检查点之后，数据库高速缓存的所有内容都在磁盘上。

数据库服务器在下列条件下对数据库执行检查点操作：

- 作为数据库关闭操作的一部分
- 自上一个检查点以来经过的时间超过 `-gc` 服务器选项的设置
- 估计的执行恢复操作的时间超过 `-gr` 服务器选项的设置
- 数据库服务器空闲的时间长得足以写入所有脏页
- 某些 DDL 语句（例如 ALTER TABLE、DROP TABLE、DROP INDEX、LOAD TABLE 或 BACKUP）已执行
- 连接发出 CHECKPOINT 语句
- 数据库服务器在没有使用事务日志的情况下运行时一个事务被提交

要确保拥有检查点之间的所有已提交事务的完整副本，需要数据库文件和事务日志。

另请参见

- “了解检查点日志”一节第 17 页
- “数据库服务器如何确定何时进行检查点操作”一节第 853 页
- “-gc 服务器选项”一节第 178 页
- “-gr 服务器选项”一节第 184 页

数据库服务器如何确定何时进行检查点操作

自上一个检查点以来经过的时间越长、工作量越大，将脏页写入磁盘的优先级就越高。该优先级由以下因素确定：

- **检查点紧急情况** 上一检查点与当前时间之间的时间间隔，显示为数据库检查点时间设置的百分比。使用 `-gc` 服务器选项或 `checkpoint_time` 数据库选项可以设置检查点之间的最大时间间隔（以分钟为单位）。如果指定了 `-gc`，则数据库中的 `checkpoint_time` 选项设置被忽略。
- **恢复紧急情况** 一种启发式算法，用于估计在数据库失败时立即恢复它所需的时间。使用 `-gr` 服务器选项或 `recovery_time` 数据库选项可以设置在系统出现故障时进行恢复的最长时间（以分钟为单位）。如果指定了 `-gr`，则数据库中的 `recovery_time` 选项设置被忽略。

只有当数据库服务器没有足够的空闲时间写入脏页时，检查点和恢复紧急情况值才是重要的。根据 `recovery_time` 和 `checkpoint_time` 选项的组合确定检查点之间时间间隔的下限。在立即强迫检查点的情况下不考虑 `recovery_time` 选项设置。

检查点越频繁，恢复越快，但是服务器写出脏页的工作量会越大。

如果脏页数量因数据库中的其它活动而下降到零，并且检查点紧急程度为 33% 或更高，则会自动执行检查点操作（因为这时便于执行此操作）。

检查点紧急情况值和恢复紧急情况值会一直增加，直到出现检查点，此时这些值会降至零。

另请参见

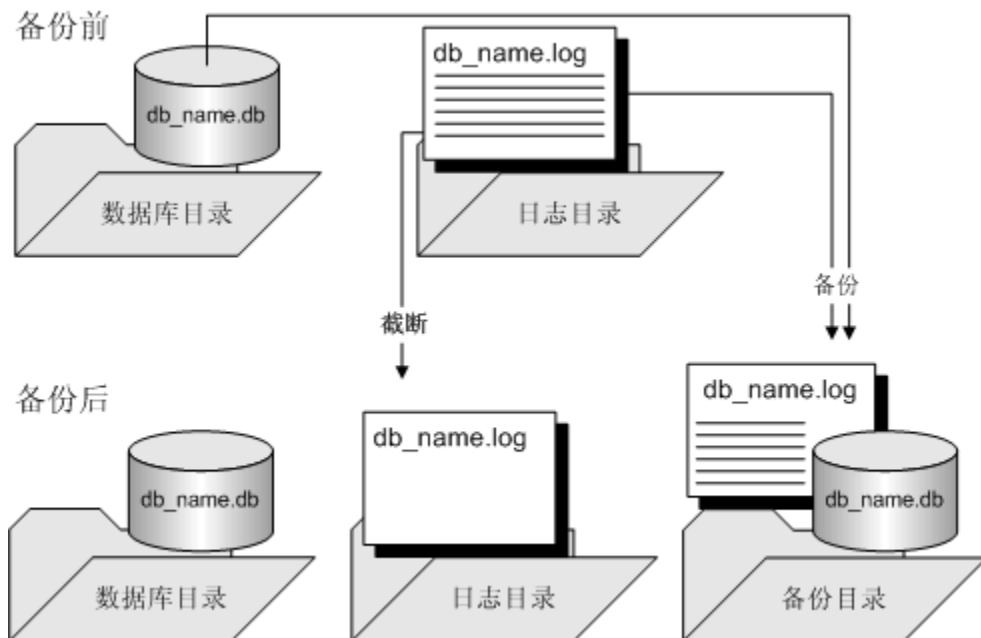
- “了解检查点日志”一节第 17 页
- “-gc 服务器选项”一节第 178 页
- “`checkpoint_time` 选项 [数据库]”一节第 488 页
- “-gr 服务器选项”一节第 184 页
- “`recovery_time` 选项 [数据库]”一节第 538 页

管理事务日志

执行备份时，在缺省情况下备份会创建事务日志的当前状态的副本，并保留事务日志不变。如果您的数据库参与了同步或复制，则可能需要在恢复数据库之后访问事务日志的旧副本。

在许多情况下，如果存在磁盘空间限制，则让事务日志无限增大是不切实际的。要释放磁盘空间，可以选择在备份完成时删除事务日志的内容来进行释放。如果数据库参与了复制，则不要选择此选项，因为复制时需要访问事务日志。

完全备份会截断日志文件，下图将对其进行阐释。在增量备份中，只备份事务日志。



如果在每次增量备份之后都删除事务日志，则会使从数据库文件的介质故障中进行恢复变成一项更加复杂的任务。必须按顺序应用每个事务日志以使数据库更新到最新状态，并且自上次完全备份以来可能会有多个不同的事务日志。

由于 MobiLink 不依赖于事务日志，所以可以对作为 MobiLink 统一数据库运行的数据库使用这种备份。如果要运行 SQL Remote 或 MobiLink *dbmsync.exe* 应用程序，则必须使用可保存旧事务日志的模式。

请参见：

- “执行备份和重命名原始事务日志”一节第 848 页
- “在备份过程中重命名事务日志的备份副本”一节第 849 页
- “执行备份和删除原始事务日志”一节第 850 页

脱机事务日志

除了备份事务日志以外，备份操作还将联机事务日志的文件名重命名为 *YYMMDDxx.log* 形式。该文件不再由数据库服务器使用，但是可由 [消息代理] 和 [复制代理] 使用。它被称作**脱机**事务日志。新的联机事务日志在启动时将与旧联机事务日志同名。

YYMMDDxx.log 文件名可用于区分文件，而不用于排序。例如，如果首次备份是在 2000 年 12 月 10 日，则重命名的日志文件名为 *001210AA.log*。前两位数字表示年份，接下来两位数字表示月份，再往后两位数字表示该月的第几天，最后两个字符区分在同一天进行的不同备份。

[消息代理] 和 [复制代理] 可以使用脱机副本来根据需要提供旧事务。如果您将 `delete_old_logs` 数据库选项设置为 On，则 [消息代理] 和 [复制代理] 在不再需要脱机文件时删除它们以节省磁盘空间。

回退日志

在更改数据库的内容时保留**回退日志**的目的在于，如果事务回退或者在系统出现故障时有未提交的事务，该日志可用于取消更改。每个连接都有一个单独的回退日志。当提交或回退了事务时，该连接的回退日志的内容将被删除。回退日志存储在数据库中，回退日志页与发生更改的其它页一起复制到检查点日志中。

回退日志还称作**撤消日志**。

有关事务处理的详细信息，请参见“[使用事务和隔离级别](#)”《[SQL Anywhere 服务器 - SQL 的用法](#)》。

了解并行数据库备份

在通过指定 `-s` 选项使用备份实用程序 (`dbbackup`) 或通过使用 `BACKUP DATABASE` 语句执行服务器端映像备份时，可以执行并行数据库备份。并行备份使用物理设备级并行操作来减少完成备份操作所需的全部时间。Windows Mobile 不支持并行备份。

数据库服务器为每个存储数据库文件的驱动器创建读取程序线程。写入程序线程是为备份目录所在的目标驱动器而创建的。使用单独的读取程序和写入程序允许 I/O 操作以并行而非顺序的方式执行。

并行备份的性能受系统中速度最慢的组件所限。通常是物理磁盘，但也可能是其它组件，如 I/O 控制器或系统总线。这些组件中的每一个都有一个最大的数据传输率。

`BACKUP DATABASE` 语句和备份实用程序 (`dbbackup`) 为您提供了一些可配置并行备份行为的选项，包括：

- 复制检查点日志的时间和方式
- 将数据从数据库服务器一次性传送到 `dbbackup` 时使用的最多页数（仅在使用 `dbbackup` 时可用）
- 添加多个写入程序（仅用于 `BACKUP` 语句）

应始终在单独的物理驱动器上进行备份。这可以通过 I/O 并行操作改进性能，并且在硬件发生故障时还会提高数据的安全性。

另请参见

- “[BACKUP 语句](#)”一节《[SQL Anywhere 服务器 - SQL 参考](#)》
- “[备份实用程序 \(dbbackup\)](#)”一节第 698 页

校验数据库

目录

校验简介	858
使用校验和检测损坏	859
在校验数据库时提高性能	862

校验简介

在数据库服务器尝试访问数据库的受影响部分之前，数据库文件的损坏情况可能并未得到确认。应定期使用 Sybase Central 中的 [\[校验数据库向导\]](#) 或校验实用程序 (dbvalid) 等工具检查数据库是否有效。必须拥有 VALIDATE 权限才能执行校验活动。请参见 [“VALIDATE 特权”一节第 427 页](#)。

根据指定的选项，校验可以包括校验和、索引数据的正确性，以及数据库中的所有表页是否属于对象。快速数据库校验 (-fx 选项) 不校验数据、连续行结构或外键关系。

校验要求以独占方式访问进行校验的对象。因此，最好在数据库中没有其它活动时进行校验。如果您能够确定在备份过程中未执行任何事务，则数据库服务器无需执行恢复步骤。在这种情况下，您可以使用只读数据库选项对备份执行有效性检查。请参见 [“-r 服务器选项”一节第 204 页](#)。

提示

使用 BACKUP 语句时结合 WAIT BEFORE START 子句可确保在开始备份时没有任何事务在执行。

如果数据库文件中的基表损坏，则应将这种情况视为介质故障，并从以前的备份中恢复。如果索引损坏，最好先卸载没有索引的数据库，然后重装。

另请参见

- [“校验数据库”一节第 859 页](#)
- [“校验事务日志”一节第 851 页](#)
- [“校验表”一节第 860 页](#)
- [“VALIDATE 语句”一节 《SQL Anywhere 服务器 - SQL 参考》](#)
- [“在校验数据库时提高性能”一节第 862 页](#)

使用校验和检测损坏

校验和用于确定数据库页是否已在磁盘上进行了修改。在您创建启用校验和的数据库时，在将每一页写入磁盘前为其计算校验和。下次从磁盘中读取该页时，就会重新计算该页的校验和，并将其与该页上存储的校验和进行比较。如果校验和不同，说明页面在磁盘中已被修改，并发生错误。

可通过执行以下语句来检查创建数据库时是否启用了校验和：

```
SELECT DB_PROPERTY ( 'Checksum' );
```

如果开启了校验和，则此查询将返回 ON，否则将返回 OFF。

校验校验和

如果在启用了校验和的情况下创建数据库，则可以检查磁盘页的有效性。校验和校验需要 DBA 或 VALIDATE 权限。

对于启用了校验和的数据库，会为每个数据库页计算校验和，并在该页写入磁盘时存储该值。可以使用 Sybase Central 中的校验实用程序 (dbvalid) 或 [\[校验数据库向导\]](#) 执行校验和校验，这包括从磁盘读取数据库页并计算页面的校验和。如果计算出的校验和与页面的存储校验和不匹配，则说明该页面在磁盘上时或写入页面时可能已被修改或者已损坏。如果有一个或多个页面已损坏，则会返回错误，并在数据库服务器消息窗口中显示有关该无效页的信息。

有关校验和校验的详细信息，请参见“[VALIDATE 语句](#)”一节 [《SQL Anywhere 服务器 - SQL 参考》](#)和“[校验实用程序 \(dbvalid\)](#)”一节第 810 页。

自动创建校验和

在以下情况中，将为数据库启用校验和，而无论在创建数据库时指定了怎样的校验和设置：

- **重要页** 无论是否启用校验和，数据库服务器都会计算所有数据库中重要数据库页面的校验和。这些校验和用于检测脱机损坏情况，这有助于减少由于重要页面发生错误而损坏其它数据的可能。因为数据库服务器会计算这些校验和，因此，如果没有启用校验和的数据库损坏，则数据库服务器会因出现致命错误而关闭。
同样，如果您校验到数据库没有启用校验和，但有一个错误的重要页面，则 dbvalid 仍可以返回有关校验和违规的警告。
- **Windows Mobile 数据库** 数据库服务器自动为在 Windows Mobile 上运行的数据库启用校验和，以帮助尽早地检测到数据库文件是否损坏。
- **在某些存储介质上运行的数据库** 当数据库在可能不太安全的存储介质（例如网络或可移除驱动器）上运行时，数据库服务器会自动为数据库启用校验和。只要数据库保存在这样一台设备上校验和就保持启用状态，并在页面写入时对其进行校验和检查。如果数据库移动到更可靠的存储设备上，则当已校验和的页进入数据库服务器高速缓存时数据库服务器会校验这些页的校验和。

校验数据库

您必须具有 DBA 或 VALIDATE 权限才能校验数据库。

小心

应在没有任何连接对数据库进行更改时对表或整个数据库进行校验；否则，可能会报告错误，指出某种形式的数据库损坏，而实际上并没有任何损坏。

◆ **检查整个数据库的有效性 (Sybase Central):**

1. 以具有 DBA 权限的用户身份连接到数据库。
2. 在左窗格中，选择数据库。
3. 从 [文件] 菜单中选择 [校验数据库]。
4. 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]:

- 右击数据库并选择 [校验数据库]。
- 选择数据库，然后选择 [工具] » [SQL Anywhere 11] » [校验数据库]。

◆ **检查整个数据库的有效性 (SQL):**

- 执行 sa_validate 存储过程:

```
CALL sa_validate;
```

此过程返回一列，其名称为 Messages。如果所有表都有效，则此列包含 [未检测到错误]。

有关详细信息，请参见“sa_validate 系统过程”一节《SQL Anywhere 服务器 - SQL 参考》。

◆ **检查整个数据库的有效性 (命令行)**

- 运行 dbvalid 实用程序:

```
dbvalid -c "connection-string"
```

请参见“校验实用程序 (dbvalid)”一节第 810 页。

注意

如果要检查备份副本的有效性，应在只读模式下运行数据库，以使其不会以任何方式被修改。只有当备份过程中未执行任何事务时才可以这样做。请参见“-r 服务器选项”一节第 204 页。

校验表

您必须具有 DBA 或 VALIDATE 权限才能校验表。

◆ 检查表的有效性 (Sybase Central):

1. 以具有 DBA 权限的用户身份连接到数据库。
2. 在左窗格中，双击 [表]。
3. 右击表并选择 [校验]。
4. 单击 [确定]。

◆ 检查表的有效性 (SQL):

- 执行 VALIDATE TABLE 语句:

```
VALIDATE TABLE table-name;
```

注意

- 如果报告了错误，则可以删除表的所有索引和键，然后再重新创建它们。还需要重新创建表的外键。
- 如果您对某个特定索引有所怀疑，则可以执行 ALTER INDEX ...REBUILD 语句来重建受损索引。请参见“ALTER INDEX 语句”一节《SQL Anywhere 服务器 - SQL 参考》。
- 解决由 VALIDATE TABLE 报告的错误的另一个解决方案是卸载并重装整个数据库。应使用 dbunload 的 -u 选项，这样卸载进程就不会尝试使用可能已损坏的索引来对数据进行排序。

在校验数据库时提高性能

如果在运行于服务器上的大型数据库上使用 `VALIDATE TABLE` 语句，而服务器的高速缓存又太小，无法包含表及其最大索引，该语句就会很慢。通常，对于每个索引都至少会读取一次表中的所有页。而且，如果索引查寻需要进行完全比较，则可以读取的页数与表中的行数（而非页数）成比例。

如果您希望减少校验所需的时间，则可以将 `WITH EXPRESS CHECK` 选项与 `VALIDATE TABLE` 语句一起使用，或者将 `-fx` 选项与 `dbvalid` 实用程序一起使用。根据数据库的大小、高速缓存的大小和所需的校验类型，这两个功能可显著减少执行校验所需的时间。

快速校验会导致读取表的每一行并评估所有列。对每个索引都完全扫描一次并完成检查，以确保索引中引用的行存在于表中。快速检查选项还对单个索引页的有效性进行检查。表中的行数必须与索引中的条目数相匹配。快速选项不是对每一行都执行单个索引查寻，因此会节省时间。

因为快速检查功能不执行单个查寻，所以快速校验功能有可能（尽管不太可能）忽略某种形式的索引损坏。如果发生索引损坏，由于校验已经确认所有数据可被读取，所以，通过卸载并重建数据库可恢复数据。也可以使用 `ALTER INDEX` 语句的 `REBUILD` 子句修正索引损坏。请参见“[ALTER INDEX 语句](#)”一节《[SQL Anywhere 服务器 - SQL 参考](#)》。

- “[VALIDATE 语句](#)”一节《[SQL Anywhere 服务器 - SQL 参考](#)》
- “[校验实用程序 \(dbvalid\)](#)”一节第 810 页
- “[sa_validate 系统过程](#)”一节《[SQL Anywhere 服务器 - SQL 参考](#)》

使用调度和事件自动完成任务

目录

调度与事件使用简介	864
了解事件	865
了解调度	866
了解系统事件	868
了解事件处理程序	871
调度和事件内部	873
事件处理任务	875

调度与事件使用简介

很多数据库管理任务最好是有计划有步骤地完成。例如，定期备份过程是适当的数据库管理过程的重要组成部分。

在 SQL Anywhere 中，可以通过将某个**事件**添加到数据库中并提供该事件的调度，来自动完成例行的任务。每当调度中的某个时间经过时，数据库服务器都会执行被称为**事件处理程序**的一系列操作。

数据库管理还包括在出现某些情况时采取相应的操作。例如，当事务日志所在的磁盘快写满时，可能应该用电子邮件通知系统管理员，以便管理员进行处理。通过为一组**系统事件**中的某个事件定义事件处理程序，也可以自动完成这些任务。

问题与解答

要回答此问题……	请考虑阅读……
什么是调度？	“了解调度”一节第 866 页
什么是事件？	“了解事件”一节第 865 页
什么是系统事件？	“了解系统事件”一节第 868 页
什么是事件处理程序？	“了解事件处理程序”一节第 871 页
如何调试事件处理程序？	“开发事件处理程序”一节第 871 页
数据库服务器如何使用调度来触发事件处理程序？	“数据库服务器如何检查调度事件”一节第 873 页
如何调度定期备份？	“了解调度”一节第 866 页
数据库服务器可以使用哪种系统事件来触发事件处理程序？	“了解系统事件”一节第 868 页 “CREATE EVENT 语句”一节 《SQL Anywhere 服务器 - SQL 参考》
在哪个连接上执行事件处理程序？	“事件处理程序是如何执行的”一节第 874 页
事件处理程序如何获得有关触发它们的事件的信息？	“开发事件处理程序”一节第 871 页 “EVENT_PARAMETER 函数 [System]”一节 《SQL Anywhere 服务器 - SQL 参考》

了解事件

在 SQL Anywhere 中，可以通过将某个事件添加到数据库中并提供该事件的调度，来自动完成例行的任务。SQL Anywhere 支持三类事件：

- **调度事件** 具有相关联的调度并在指定时间执行。请参见“[了解调度](#)”一节第 866 页。
- **系统事件** 与由数据库服务器跟踪的特殊条件类型关联。请参见“[了解系统事件](#)”一节第 868 页。
- **手工事件** 通过使用 TRIGGER EVENT 语句显式地触发。请参见“[触发事件处理程序](#)”一节第 876 页。

每次执行事件处理程序后，如果没有出现错误便会发生 COMMIT。如果出现错误，则会发生 ROLLBACK。

了解调度

通过对活动进行调度，可以确保在一组预置的时间执行一组操作。调度信息和事件处理程序都存储在数据库本身中。

可以通过将多个调度与一个指定的事件相关联，来定义复杂的调度，但通常并不必要这样做。例如，一家零售店可能希望某事件在营业时间每小时发生一次，而营业时间根据一周中的不同日期而变化。通过定义各自拥有自己的调度的多个事件，以及调用一个公用存储过程，您可以达到同样的效果。

对事件进行调度时，可以使用完整的英文日期名称（Monday、Tuesday 等），也可以使用日期的简写形式（Mon、Tue 等）。注意，如果要想以英文以外的其它语言运行的服务器能够识别日期名称，您必须使用完整的英文日期名称。

以下示例可以帮助您初步了解可能有用的调度操作。

示例

每天凌晨 1 点执行增量备份：

```
CREATE EVENT IncrementalBackup
SCHEDULE
  START TIME '1:00 AM' EVERY 24 HOURS
HANDLER
BEGIN
  BACKUP DATABASE DIRECTORY 'c:\\backup'
  TRANSACTION LOG ONLY
  TRANSACTION LOG RENAME MATCH
END;
```

在每个工作日结束时汇总订单：

```
CREATE EVENT Summarize
SCHEDULE
  START TIME '6:00 pm'
  ON ( 'Monday', 'Tuesday', 'Wednesday', 'Thursday',
      'Friday' )
HANDLER
BEGIN
  INSERT INTO OrderSummary
  SELECT CURRENT DATE,
    COUNT( * ),
    SUM( amount )
  FROM Orders
  WHERE date_ordered = current date
END;
```

另请参见

- “CREATE EVENT 语句”一节 《SQL Anywhere 服务器 - SQL 参考》

定义调度

为灵活起见，调度定义具有下列几个组成部分：

- **名称** 每个调度定义都有一个名称。您可以将多个调度指派给某个特定事件，这在设计复杂调度时非常有用。
- **开始时间** 您可以为事件定义开始时间，即开始执行事件的时间。
- **范围** 如果不指定开始时间，可以指定事件处于活动状态的时间范围。事件在指定的开始和结束日期范围内发生。频率由指定的周期决定。
- **定期** 每个调度都可以反复循环。事件是按一定频率触发的，可以按某些日期的点钟、分钟或秒指定频率，而这些日期可以指定为星期几或几号。反复出现的事件包含一个 **EVERY** 或 **ON** 子句。

可使用 CREATE EVENT 语句或 [创建调度向导] 为事件定义调度。

有关在创建事件时添加调度的信息，请参见“[CREATE EVENT 语句](#)”一节《[SQL Anywhere 服务器 - SQL 参考](#)》。

◆ 为事件创建调度 (Sybase Central)

1. 以具有 DBA 权限的用户身份连接到数据库。
2. 双击 [事件]。
3. 双击要为其创建调度的事件。
4. 单击 [调度] 选项卡。
5. 从 [文件] 菜单选择 [新建] » [调度]。
6. 请按照 [创建调度向导] 中的说明进行操作。

了解系统事件

SQL Anywhere 跟踪着多个系统事件。每个系统事件都提供一个您可以在其上挂接一组操作的挂钩。数据库服务器为您跟踪事件，并在系统事件满足提供的**触发条件**时执行操作（操作在事件处理程序中定义）。

有关触发条件的详细信息，请参见“[定义事件的触发条件](#)”一节第 869 页。

通过将事件处理程序定义为在发生选定系统事件且该事件符合定义的触发条件时执行，可以提高数据的安全性并有助于简化管理。如果在执行期间未检测到任何错误，则将提交事件处理程序的操作。如果检测到错误，将回退。

可用的系统事件如下：

- **BackupEnd** 您可以使用 BackupEnd 事件类型在备份结束时执行操作。
- **连接事件** 在建立连接时 (Connect) 或连接尝试失败时 (ConnectFailed)。为安全起见，您可能希望使用这些事件。作为连接事件处理程序的替代方法，可能要考虑使用登录过程。请参见“[login_procedure 选项 \[数据库\]](#)”一节第 513 页。
- **DatabaseStart** 您可以使用 DatabaseStart 事件类型在启动数据库时执行操作。
- **死锁** 您可以使用 Deadlock 事件在发生死锁时执行操作。事件处理程序可以使用 sa_report_deadlocks 过程以获得有关导致死锁的条件的信息。使用 Deadlock 事件时，应配置数据库服务器以便通过将 log_deadlocks 选项设置为 On 来捕获死锁信息，并且通过使用 sa_server_option 或 -zl 服务器选项来启用 RememberLastStatement 功能。

连接死锁和线程死锁触发 Deadlock 事件。Deadlock 事件仅能提供可通过 sa_report_deadlocks 系统过程获得的信息。但是，通过使用此事件可及时地对死锁发生作用。由于数据库服务器能够保留的与死锁相关的信息量是有限的，所以快速响应会非常重要。请参见：

 - “[sa_report_deadlocks 系统过程](#)”一节《[SQL Anywhere 服务器 - SQL 参考](#)》
 - “[log_deadlocks 选项 \[数据库\]](#)”一节第 511 页
 - “[死锁](#)”一节《[SQL Anywhere 服务器 - SQL 的用法](#)》
- **断开连接** 您可以使用 Disconnect 事件在用户或应用程序断开连接时执行操作。
- **可用磁盘空间** 跟踪数据库文件 (DBDiskSpace)、日志文件 (LogDiskSpace) 或临时文件 (TempDiskSpace) 所在设备的可用磁盘空间。此系统事件在 Windows Mobile 上不可用。

在磁盘空间不足时，您可能希望使用磁盘空间事件来提醒管理员。

您可以在启动数据库服务器时指定 -fc 选项，以在数据库服务器出现文件系统已满时执行回调函数。请参见“[-fc 服务器选项](#)”一节第 175 页。
- **文件大小** 文件达到指定的大小。可以将此事件用于数据库文件 (GrowDB)、事务日志 (GrowLog) 或临时文件 (GrowTemp)。

您可能希望使用文件大小事件来跟踪对数据库的异常操作或监控批量操作。
- **GlobalAutoIncrement** 当用 GLOBAL AUTOINCREMENT 定义的列的剩余值数目不足其范围的百分之一时，就会触发 GlobalAutoIncrement 事件。基于表和提供给此事件作为参数的剩余值数目，可使用此事件为 global_database_id 选项请求新值。要获得事件中表的其余值，请使用具有 RemainingValues 参数和 TableName 参数的 EVENT_PARAMETER 函数。RemainingValues 将

会返回可为该列生成的剩余值的数量，而 `TableName` 则会返回含有 GLOBAL AUTOINCREMENT 列（此列接近其范围末尾）的表。请参见“[EVENT_PARAMETER 函数 \[System\]](#)”一节《[SQL Anywhere 服务器 - SQL 参考](#)》。

- **RAISERROR 错误** 当执行一个 RAISERROR 语句时，您可以使用 RAISERROR 事件类型来执行操作。通过使用 `EVENT_CONDITION` 函数，可在事件处理程序中确定在 RAISERROR 语句中使用的错误号（例如，`EVENT_CONDITION('ErrorNumber')`）。
- **空闲时间** 数据库服务器已经空闲了指定的时间 (`ServerIdle`)。您可能希望使用此事件类型在安静期间执行例行的维护操作。
- **数据库镜像** 当主服务器与镜像服务器或仲裁服务器之间的连接中断时，将会触发 `MirrorServerDisconnect` 事件。要获得断开连接的服务器的名称，请使用具有 `MirrorServerName` 参数的 `EVENT_PARAMETER` 函数。请参见“[EVENT_PARAMETER 函数 \[System\]](#)”一节《[SQL Anywhere 服务器 - SQL 参考](#)》。

只要服务器具有数据库的所有权，便会触发 `MirrorFailover` 事件。例如，在服务器初次启动并确定其将会具有数据库时，便会触发该事件。当以前起镜像作用的服务器确定主服务器已关闭，并在与仲裁服务器协商之后确定其应拥有所有权时，也会触发该事件。

当前正在充当镜像服务器的服务器不会触发事件，因为其数据库备份仍处于启动状态。同样，不能将镜像事件定义为在仲裁服务器上执行，因为事件仅在定义它们的数据库环境中运行，并且仲裁服务器也不会使用要被镜像的数据库副本。请参见“[数据库镜像系统事件](#)”一节第 901 页。

定义事件的触发条件

每个事件定义都有一个与其关联的系统事件。它也有一个或多个触发条件。在满足系统事件的触发条件时，会触发事件处理程序。

触发条件包括在 `CREATE EVENT` 语句的 `WHERE` 子句中，可以使用 `AND` 关键字对触发条件进行组合。每个触发条件的形式如下：

event_condition(condition-name) comparison-operator value

condition-name 参数是一组预置字符串中的一个字符串，这些字符串适用于不同的事件类型。例如，您可以使用 `DBSize`（以兆字节为单位的数据库文件大小）来构建适合于 **GrowDB** 系统事件的触发条件。数据库服务器不检查条件名是否与事件类型匹配：您应负责确保该条件在事件类型上下文中有意义。

示例

- 将事务日志的大小限制到 10 MB:

```
CREATE EVENT LogLimit
TYPE GrowLog
WHERE event_condition( 'LogSize' ) > 10
HANDLER
BEGIN
  IF EVENT_PARAMETER( 'NumActive' ) = 1 THEN
    BACKUP DATABASE
    DIRECTORY 'c:\\logs'
    TRANSACTION LOG ONLY
    TRANSACTION LOG RENAME MATCH;
```

```
    END IF;  
END;
```

- 当数据库文件所在设备上的可用磁盘空间低于 10% 时，通知管理员，但执行处理程序的频率不超过每五分钟（即 300 秒）一次：

```
CREATE EVENT LowDBSpace  
TYPE DBDiskSpace  
WHERE event_condition( 'DBFreePercent' ) < 10  
AND event_condition( 'Interval' ) >= 300  
HANDLER  
BEGIN  
    CALL xp_sendmail( recipient='DBAdmin',  
                    subject='Low disk space',  
                    "message"='Database free disk space '  
                    || EVENT_PARAMETER( 'DBFreeSpace' ) );  
END;
```

- 通知管理员可能有人试图入侵数据库：

```
CREATE EVENT SecurityCheck  
TYPE ConnectFailed  
HANDLER  
BEGIN  
    DECLARE num_failures INT;  
    DECLARE mins INT;  
    INSERT INTO FailedConnections( log_time )  
    VALUES ( CURRENT_TIMESTAMP );  
  
    SELECT COUNT( * ) INTO num_failures  
    FROM FailedConnections  
    WHERE log_time >= DATEADD( minute, -5,  
    current_timestamp );  
    IF( num_failures >= 3 ) THEN  
        SELECT DATEDIFF( minute, last_notification,  
        current_timestamp ) INTO mins  
        FROM Notification;  
        IF( mins > 30 ) THEN  
            UPDATE Notification  
            SET last_notification = current_timestamp;  
            CALL xp_sendmail( recipient='DBAdmin',  
                            subject='Security Check', "message"=  
                            'over 3 failed connections in last 5 minutes' )  
        END IF  
    END IF  
END;
```

- 在服务器已经空闲十分钟时运行一个进程。执行频率不要超过每小时一次：

```
CREATE EVENT Soak  
TYPE ServerIdle  
WHERE event_condition( 'IdleTime' ) >= 600  
AND event_condition( 'Interval' ) >= 3600  
HANDLER  
BEGIN  
    MESSAGE ' Insert your code here ... '  
END;
```


了解事件处理程序

事件处理程序与触发了该事件的操作在不同的连接上执行，因此不与客户端应用程序进行交互。它们是使用事件创建者的权限执行的。

开发事件处理程序

不管是用于调度事件的处理，还是用于系统事件处理，事件处理程序都包含复合语句，而且在很多方面都与存储过程类似。您可以添加循环、条件执行等，还可以使用 SQL Anywhere 调试程序来调试事件处理程序。

每次执行事件处理程序后，如果没有出现错误便会发生 COMMIT。如果出现错误，则会发生 ROLLBACK。

事件处理程序的上下文信息

与储存过程不同，事件处理程序不带任何参数。可使用 EVENT_PARAMETER 函数访问在其中触发了事件的上下文的相关信息。返回的信息包括导致事件触发的连接 ID 和用户 ID、事件名以及该事件的执行次数。请参见“[EVENT_PARAMETER 函数 \[System\]](#)”一节《[SQL Anywhere 服务器 - SQL 参考](#)》。

测试事件处理程序

在开发过程中，您希望在方便时触发事件处理程序。即使在没有出现触发条件或调度时间时，您也可以使用 TRIGGER EVENT 语句显式导致事件的执行。但是，TRIGGER EVENT 不会导致被禁用的事件处理程序执行。请参见“[TRIGGER EVENT 语句](#)”一节《[SQL Anywhere 服务器 - SQL 参考](#)》。

尽管在生产数据库上开发事件处理程序不是很好的做法，但您可以从 Sybase Central 中，或显式使用 ALTER EVENT 语句来禁用事件处理程序。

代码共享

用同一组操作来处理多个事件是很有用的。例如，如果数据库文件或日志文件所在设备上磁盘空间有限，您可能会需要执行通知操作。为此，创建一个存储过程并在每个事件处理程序的正文中调用它，从而将任何所需的上下文信息作为参数传递给该过程。

调试事件处理程序

调试事件处理程序与调试存储过程非常类似。事件处理程序出现在事件列表中。

有关详细信息和步骤说明，请参见“[调试事件处理程序](#)”一节第 877 页。

隐藏事件处理程序

可以使用 SET HIDDEN 子句来隐藏某个事件处理程序的定义。指定 SET HIDDEN 子句的结果是永久模糊处理存储在 ISYSEVENT 系统表的操作列中的事件处理程序定义。请参见“[ALTER EVENT 语句](#)”一节《[SQL Anywhere 服务器 - SQL 参考](#)》。

限制活动事件

通过使用 NumActive 事件参数，您还可以确定某个特定事件处理程序有多少个实例当前处于活动状态。如果您要限制事件处理程序以便在给定的时间只执行一个实例，则此函数是很有用的。

有关 NumActive 事件参数的详细信息，请参见“[EVENT_PARAMETER 函数 \[System\]](#)”一节《[SQL Anywhere 服务器 - SQL 参考](#)》。

调度和事件内部

本节介绍数据库服务器如何处理调度和事件定义。

数据库服务器如何检查系统事件

系统事件是按照其**事件类型**进行分类的，事件类型在 CREATE EVENT 语句中直接指定或者使用 Sybase Central 指定。有两种事件类型：

- **活动的事件类型** 有些事件类型是数据库服务器本身所执行的操作的结果。这些活动的事件类型包括增长的数据库文件或不同的数据库操作或 RAISERROR 的开始和结束（BackupEnd 等）。
当数据库服务器采取操作时，它会检查是否满足在 WHERE 子句中定义的触发条件，如果满足条件，则触发为该事件类型定义的任何事件。
- **轮询的事件类型** 有些事件类型（例如，DBDiskSpace 等可用磁盘空间类型以及 IdleTime 类型）并不仅仅是由数据库操作触发的。

对于这些事件类型，数据库服务器每隔三十秒轮询一次，大约在启动数据库服务器三十秒后开始进行轮询。

对于 IdleTime 事件类型，数据库服务器会检查服务器是否已经空闲了整三十秒。如果没有启动请求且当前没有请求处于活动状态，则服务器会将空闲检查间隔时间（以秒为单位）与总空闲时间累加；否则，将总空闲时间重置为 0。因此 IdleTime 的值始终是三十秒的倍数。当 IdleTime 大于在触发条件中指定的间隔时，会触发与 IdleTime 关联的事件处理程序。

数据库服务器如何检查调度事件

在数据库服务器启动时以及每当调度的事件处理程序完成时，都会进行调度的事件时间的计算。

根据在调度定义中指定的增量，将增量与上一开始时间累加起来计算下一个调度时间。如果执行事件处理程序花费的时间比指定的增量大，导致下一时间早于当前时间，则数据库服务器会按指定的增量计算下一个调度时间，直到下一个调度时间是将来的时间。

例如，其执行需要六十五分钟且被请求在 9:00 和 5:00 之间每小时运行一次的事件处理程序，将每隔两小时运行一次，即在 9:00、11:00、1:00 等时间运行。

为运行一个在 9:00 和 5:00 之间工作并在下一次执行之前延迟一段时间的进程，您可以将处理程序定义为在其完成时间超过之前不断循环，并且每次迭代之间都有一个 WAITFOR 语句。

如果您正在间歇地运行数据库服务器，而且不是在调度时间运行，则在启动时事件处理程序不会运行。而是会在启动时计算下一个调度时间。例如，如果您将一个备份预定在每天凌晨一点发生，但在每个工作日结束时都会关闭数据库服务器，则该备份永远也不会发生。

如果一个事件的下一调度执行时间是在一小时以后，则数据库服务器将以小时为基础重新计算它的下一个调度时间。这允许事件在系统时钟调整（系统时钟更改为 Daylight Saving Time 或者由 Daylight Saving Time 更改为其它）时按预期触发。

事件处理程序是如何执行的

当触发事件处理程序时，会建立一个可在其上执行事件处理程序的临时内部连接。处理程序不是在导致处理程序触发的连接上执行的，因此诸如 `MESSAGE ...TO CLIENT` 之类与客户端应用程序进行交互的语句在事件处理程序内是没有意义的。同样，也不允许会返回结果集的语句。

用于执行处理程序的临时连接不计入授权的连接限制，不会为 `login_procedure` 选项指定的过程执行事件连接。

创建事件需要有 `DBA` 权限，而执行事件则需要有其创建者的权限。如果您希望事件处理程序以非 `DBA` 权限执行，则可以从处理程序内调用过程，因为存储过程以其创建者的权限运行。

任何事件错误都会记录到数据库服务器消息日志中。

事件处理程序和错误

如果在执行期间未检测到任何错误，则将提交事件处理程序中的事务。如果检测到错误，将回退。

如果错误出现在原子复合语句内且该语句具有可处理该错误的异常处理程序，则在该语句中所进行的任何更改均将保持未完成状态。如果异常处理程序不处理错误或导致另一错误（包括通过 `RESIGNAL`），则在原子复合语句内所做的更改会被撤消。

事件处理任务

本节汇总了与使用事件自动完成任务有关的说明。

将事件添加到数据库中

可以从 Sybase Central 或使用 SQL 添加事件。

◆ 将事件添加到数据库中 (Sybase Central)

1. 以具有 DBA 权限的用户身份连接到数据库。
2. 在左窗格中，右击 [事件] 并选择 [新建] » [事件]。
3. 请按照 [创建事件向导] 中的说明进行操作。
在其它任务中将详细说明各事件选项。

◆ 将事件添加到数据库中 (SQL)

1. 以具有 DBA 权限的用户身份连接到数据库。
2. 执行 CREATE EVENT 语句。

CREATE EVENT 语句包含许多选项，具体选项取决于您想要创建的事件。在其它任务中将详细说明这些选项。

请参见“CREATE EVENT 语句”一节 《SQL Anywhere 服务器 - SQL 参考》。

将手工触发的事件添加到数据库中

如果您创建的事件处理程序没有可触发它的调度或系统事件，则仅当手工触发时它才会执行。

◆ 将手工触发的事件添加到数据库中 (Sybase Central):

1. 以具有 DBA 权限的用户身份连接到数据库。
2. 在左窗格中，右击 [事件] 并选择 [新建] » [事件]。
3. 在 [您要为新事件指定什么名称] 字段中键入事件的名称，然后单击 [下一步]。
4. 选择 [手工]，然后单击 [下一步]。
5. 选择 [启用此事件] 并选择 [在所有数据库中执行]，然后单击 [下一步]。
6. 键入说明事件的注释，然后单击 [完成]。
7. 在 [SQL] 窗格中，键入事件的 SQL 语句。
8. 从 [文件] 菜单中选择 [保存]。

◆ **将手工触发的事件添加到数据库中 (SQL):**

1. 以具有 DBA 权限的用户身份连接到数据库。
2. 执行没有调度或 WHERE 子句的 CREATE EVENT 语句。CREATE EVENT 的限制语法如下:

```
CREATE EVENT event-name  
HANDLER  
BEGIN  
... //event handler  
END
```

如果您在开发事件处理程序，则以后可使用 Sybase Central 或 ALTER EVENT 语句添加调度或系统事件，以便控制事件的触发。

另请参见

- “触发事件处理程序”一节第 876 页
- “ALTER EVENT 语句”一节 《SQL Anywhere 服务器 - SQL 参考》

触发事件处理程序

除了由于调度或系统事件而执行事件处理程序外，还可以手工触发事件处理程序。在开发事件处理程序过程中，手工触发事件是很有用的；对于生产环境中的某些事件，它也是很有用的。例如，您已经有了一个按月调度的销售报表，但有时希望获得销售报表并不是由于到了月末这一原因。

有关开发事件处理程序的详细信息，请参见“开发事件处理程序”一节第 871 页。

◆ **触发事件处理程序 (Sybase Central):**

1. 以具有 DBA 权限的用户身份连接到数据库。
2. 在左窗格中，双击 [事件]。
3. 右击该事件并选择 [触发器]。
必须先启用事件，然后才能将其触发。要启用某个事件，请右击该事件并选择 [已启用]。
4. 在 [参数] 字段中，键入该事件以逗号分隔的参数列表。例如：

```
parameter=value,parameter=value
```

5. 单击 [确定]。

◆ **触发事件处理程序 (SQL):**

1. 以具有 DBA 权限的用户身份连接到数据库。
2. 执行提供了事件名称的 TRIGGER EVENT 语句。例如：

```
TRIGGER EVENT sales_report_event;
```

请参见“TRIGGER EVENT 语句”一节 《SQL Anywhere 服务器 - SQL 参考》。

调试事件处理程序

调试是软件开发必不可少的一个环节。在开发过程中可以调试事件处理程序。

◆ 调试事件处理程序 (Sybase Central)

1. 以具有 DBA 权限的用户身份连接到数据库。
2. 从 [模式] 菜单中选择 [调试]。
3. 在左窗格中，双击 [事件]。
4. 双击要调试的事件。
5. 在右窗格的 [SQL] 选项卡中，按 F9 键以设置断点。
6. 从 Interactive SQL 或另一个应用程序，使用 TRIGGER EVENT 语句触发事件处理程序。
7. 执行过程将会在您设置的断点处停止。

另请参见

- “开发事件处理程序”一节第 871 页
- “调试过程、函数、触发器和事件” 《SQL Anywhere 服务器 - SQL 的用法》

隐藏事件处理程序

为提高安全性，可以使用 ALTER EVENT 语句来隐藏某个事件处理程序的定义。这样做的结果是模糊处理存储在 ISYSEVENT 系统表的操作列中的事件处理程序定义。

◆ 隐藏事件处理程序 (SQL)

1. 以具有 DBA 权限的用户身份连接到数据库。
2. 执行 ALTER EVENT *event-name* SET HIDDEN 语句，其中 *event-name* 是要为其隐藏处理程序的事件的名称。

另请参见

- “ALTER EVENT 语句”一节 《SQL Anywhere 服务器 - SQL 参考》
- “SYSEVENT 系统视图”一节 《SQL Anywhere 服务器 - SQL 参考》

SQL Anywhere 高可用性

目录

数据库镜像简介	880
教程：使用数据库镜像	887
教程：使用多个数据库共享一个仲裁服务器的数据库镜像	891
设置数据库镜像	896
使用 SQL Anywhere Veritas 聚簇服务器代理	905

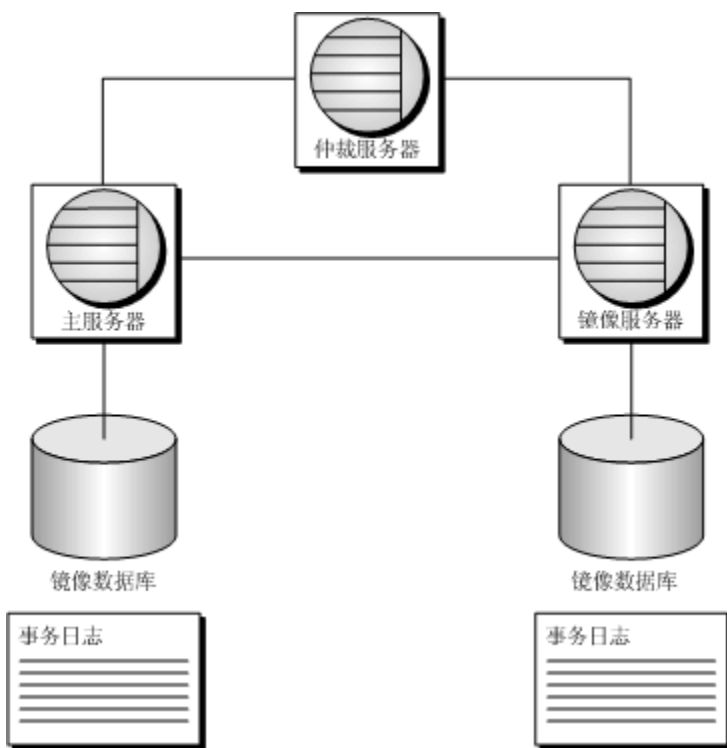
数据库镜像简介

需要单独授予许可的组成部分

数据库镜像需要单独的许可。请参见“单独授权的组件”一节《SQL Anywhere 11 - 简介》。

数据库镜像是由运行在不同计算机上的两个或三个数据库服务器组成的配置，这些服务器相互协作，以维护数据库和事务日志文件的副本。

主服务器和**镜像服务器**都用来维护数据库文件和事务日志文件的副本。在需要确定这两个服务器中的哪个服务器可以取得数据库的所有权时，则使用称为**仲裁服务器**的第三个服务器。仲裁服务器并不维护数据库的副本。由这三个数据库服务器（主服务器、镜像服务器和仲裁服务器）组成的配置称为**镜像系统**。其中，主服务器和镜像服务器合在一起称为**运行服务器**或**合作伙伴**。



客户端连接到主服务器以访问数据库。对数据库所做的任何更改都会记录在主服务器的事务日志中。提交更改后，事务日志页会发送到镜像服务器，在该处，将事务日志页应用于数据库的镜像副本。服务器充当镜像服务器时，只能以只读模式访问镜像服务器上的数据库副本。请参见“配置在镜像服务器上运行的数据库的只读访问”一节第 898 页。

当主服务器因硬件或软件故障而变为不可用时，镜像服务器便会与仲裁服务器协商以取得数据库的所有权并担当主服务器角色。为了实现所有权转让或**角色切换**，仍在工作的运行服务器和仲裁服务器必须承认镜像服务器在尝试进行角色切换时处于最新的、已同步状态。先前连接到原主服务器的所有客户端将断开，任何尚未提交的事务都会丢失。然后，客户端必须重新连接到新主服务器上的数据库才能继续访问该数据库。当原主服务器再次变为可用时，它会担当镜像服务器角色。

数据库服务器启动时会在数据库服务器消息窗口中显示状态消息，指明服务器所担当的角色以及启动进程的进度。如果因镜像系统中缺失一个或多个其它服务器而必须重新启动数据库，或者如果服务器角色从镜像服务器切换为主服务器，则会显示一条消息。

如果镜像系统中的某个服务器上发生声明失败，该服务器会将错误写入数据库服务器消息日志，然后退出。从而将它已失败通知给其它服务器，以便其它服务器采取适当的行动。

数据库镜像没有特殊的硬件或软件要求，且数据库服务器可在地理上分散的场所中运行。组成数据库镜像系统的数据库服务器既可以运行镜像数据库也可以运行非镜像数据库。同样，仲裁服务器可作为多个数据库镜像系统的仲裁服务器。

关于数据库镜像系统中每个数据库状态的详细信息存储在状态信息文件中。请参见“[状态信息文件](#)”一节第 885 页。

注意

数据库镜像不能替代备份和恢复计划。您应始终执行数据库备份和恢复策略。请参见“[数据库镜像和备份](#)”一节第 902 页和“[备份和数据恢复](#)”第 817 页。

有关在数据库镜像系统中涉及的升级 SQL Anywhere 或重建数据库的信息，请参见“[在数据库镜像系统中升级 SQL Anywhere 软件和数据库](#)”一节《[SQL Anywhere 11 - 更改和升级](#)》。

法定数目

在某个服务器能够担当主服务器角色之前，必须要达到**法定数目**，也就是说，至少有一个其它服务器必须同意该服务器可以拥有数据库。如果镜像服务器不可用，但主服务器与仲裁服务器相互连接，则主服务器会继续提供对数据库的访问。如果主服务器未达到法定数目，则不再允许访问数据库。此时，它会停止镜像数据库，尝试将其重新启动，等待重新获得法定数目，然后再使数据库可用。

启动数据库镜像系统时，数据库服务器会经历一个启动过程，首先达到法定数目，然后再接受客户端连接。以下步骤描述了这一过程的典型事件顺序：

1. 仲裁服务器等待服务器 1 和服务器 2。
2. 服务器 1 寻找仲裁服务器或服务器 2。
3. 服务器 1 连接到仲裁服务器。
4. 服务器 1 与仲裁服务器协商，以便成为主服务器。
5. 仲裁服务器和服务器 1 同意服务器 1 成为主服务器。
6. 服务器 1 开始接受连接。
7. 服务器 2 寻找服务器 1 和仲裁服务器。
8. 服务器 2 连接到仲裁服务器和服务器 1。
9. 服务器 2 请求法定数目。由于服务器 1 是主服务器，所以服务器 2 不会收到法定数目，于是它处于预备状态，等待来自服务器 1 的事务。
10. 服务器 1 向服务器 2 发送事务。

限制

使用数据库镜像时有以下限制：

- **需要网络数据库服务器** 由于镜像涉及到数据库服务器之间的网络通信，所以必须使用网络数据库服务器 (dbsrv11)；不能使用个人数据库服务器。
- **LOAD TABLE 语句** 如果在基表上执行 LOAD TABLE 语句，则必须指定 WITH ROW LOGGING 或 WITH CONTENT LOGGING 作为语句的记录级别。这些子句允许将装载的数据记录在事务日志中，以便也将数据装载到镜像数据库中。如果不指定这些语句，则报告错误。请参见“LOAD TABLE 语句”一节《SQL Anywhere 服务器 - SQL 参考》和“使用 LOAD TABLE 语句导入数据”一节《SQL Anywhere 服务器 - SQL 的用法》。
- **需要 TCP/IP** 镜像服务器之间仅允许 TCP/IP 连接。
- **故障转移和预定事件** 如果数据库有预定事件并发生了故障转移，则只要故障转移在事件的预定开始时间之前完成，该预定事件便会在镜像服务器上运行。否则，该事件的下一预定发生在镜像服务器上运行。
- **事务日志限制** 使用数据库镜像时不能截断事务日志，因为这样可能会导致事务丢失。需要时可重命名事务日志。如果要删除旧事务日志，则可以使用预定事件将其删除，只要您确定不再需要这些事务日志。例如，可创建一个事件，该事件每天都运行，用于删除存留时间超过一周的事务日志副本。请参见“数据库镜像和事务日志文件”一节第 901 页。
- **Web 服务器不能参与镜像系统** 如果 SQL Anywhere 数据库服务器包含在数据库镜像系统中，则不能将该数据库服务器用作 Web 服务器，因为发生故障转移时，该数据库服务器的 IP 地址会发生变化。

开发应用程序时的注意事项

使用数据库镜像时，在几乎所有的情况下，应用程序应该都能够以连接到非镜像数据库时的运行方式进行运行。但在开发用于数据库镜像的应用程序时还需考虑以下几点：

- 创建可重新连接到数据库的客户端（例如，发生故障转移时，用户可能需要关闭应用程序，然后再重新启动）。
- 在异步或异步满页模式下运行时，必须确定在发生故障转移且事务尚未提交到数据库时会发生什么情况。
- 镜像服务器取得数据库所有权后必须回退未完成的事务。事务越长，回退事务花费的时间就越多。故障转移的恢复速度由客户端的数量和需要回退的事务的长度来决定。如果您很关注恢复速度，就要尽可能地设计使用短事务的应用程序。

升级 SQL Anywhere

有关升级用于数据库镜像系统的 SQL Anywhere 的信息，包括应用 EBF，请参见“在数据库镜像系统中升级 SQL Anywhere 软件和数据库”一节《SQL Anywhere 11 - 更改和升级》。

数据库镜像的优点

镜像具有以下几个优点：

- 存在仲裁服务器时，从主服务器到镜像服务器的故障转移会自动进行。如果在同步模式下运行，则故障转移期间不会丢失任何已提交的事务。
- 故障转移速度很快，因为镜像服务器已经应用了事务日志。如果镜像服务器检测到主服务器发生故障，它会回退任何尚未提交的事务，然后使数据库变为可用。
- 无需安装特殊的硬件（如共享磁盘）。
- 无需安装特殊的软件（如用于聚簇的软件）。
- 无需安装特定的操作系统版本。
- 服务器不必在地理上彼此邻近。事实上，服务器彼此远离还可以防范灾难（如火灾）。
- 镜像系统中的数据库服务器还可用于运行其它数据库。

了解仲裁服务器的角色

仲裁服务器用于解决服务器之间关于哪个服务器应成为主服务器的争议。要是没有仲裁服务器，则如果服务器 A 在服务器 B 不可用时启动，服务器 A 就无法确定其数据库文件副本是否为最新内容。使用非最新文件启动数据库会导致已应用和已提交到数据库另一副本的事务丢失。另外，当两个运行服务器重新建立通信后，数据库的另一副本将不能用于镜像。

除了用于解决启动时的争议，当两个服务器之间的通信链接已断开，但这两个服务器仍在运行时，也需要仲裁服务器。如果没有仲裁服务器，这两个服务器都会认为自己应取得数据库的所有权。同样，这种情况也会造成事务丢失和数据库不兼容。有了仲裁服务器，主服务器可以验证其是否仍然拥有数据库，从而能够继续供客户端使用。如果主服务器与镜像服务器和仲裁服务器都失去了通信，则它必须关闭，然后等待其中一个服务器变为可用。

一个仲裁服务器可担任多个镜像系统的仲裁服务器。仲裁服务器还可以担当其它数据库的数据库服务器。

选择数据库镜像模式

为镜像提供了以下三种运行模式：

- 同步
- 异步
- 异步满页

同步模式是缺省设置。这些模式控制着何时以及如何如何在镜像服务器上记录事务，请使用 `-xp` 服务器选项对其进行设置。

为数据库镜像系统选择同步模式时，必须确定在发生故障转移时是恢复速度更重要还是数据状态更重要。

可能通过查询 `MirrorMode` 数据库属性的值来检查数据库镜像模式：

```
SELECT DB_PROPERTY( 'MirrorMode' );
```

同步模式

在同步模式下，可保证将已提交事务全部记录在镜像服务器上。如果主服务器发生故障，则在镜像服务器接管后不会丢失任何已提交事务。在此模式下，提交事务后，主服务器会将事务日志页发送到镜像服务器。镜像服务器在将这些日志页写入其事务日志副本后确认该传输。主服务器在收到此确认后才会回复应用程序。

使用同步模式可实现**事务安全**，因为运行服务器处于同步状态，发送到镜像服务器的更改必须在得到确认后，主服务器才能继续。

异步模式

在异步模式下，不能保证已提交事务全部记录在镜像服务器上。在此模式下，提交事务后，主服务器会将事务日志页发送到镜像服务器。但它不等待来自镜像服务器的确认就将 COMMIT 已完成回复给应用程序。如果主服务器发生故障，镜像服务器接管后可能会丢失一些已提交事务。

异步满页模式

在异步满页（或页面）模式下，COMMIT 时不发送日志页；而是在页面填满时发送日志页。这样可减少两个数据库服务器之间的通信量，从而提高主服务器的性能。如果在 `pagetimeout` 参数指定的秒数内没有将当前日志页发送到镜像服务器，那么即使该页还没有填满也要发送它。`pagetimeout` 的缺省值为 5 秒。此模式用于限制在主服务器出现故障后镜像服务器取得数据库所有权的情况下在多长时间已提交事件有可能丢失。异步满页模式隐含着异步操作，所以主服务器不等待来自镜像服务器的确认。

出于上述原因，与同步模式相比，异步和异步满页模式速度更快但可靠性却更低。在异步或异步满页模式下，从主服务器到镜像服务器的故障转移不会自动进行，因为镜像服务器中可能没有主服务器上所应用的全部已提交事务。因此，如果使用异步或异步满页模式，则在缺省情况下，主服务器发生故障时镜像服务器不能取得数据库的所有权。如果想要在这种情况下进行自动故障转移（尽管很可能会丢失事务），请使用 `-xp` 服务器选项将 `autofailover` 选项设置为 `yes`。否则，当重新启动发生故障的服务器时，它会检测是否丢失了事务。如果丢失了事务，则将一条消息写入数据库服务器消息日志，然后关闭数据库。然后必须使用备份替换当前数据库和事务日志，镜像才能继续进行。

有关在异步或异步满页模式下，在服务器出现故障后恢复服务器的信息，请参见“[从主服务器故障恢复](#)”一节第 900 页。

注意

建议您在使用异步或异步满页模式时，将 `-xp autofailover` 选项设置为 `yes`。这样，如果主服务器出现故障，镜像服务器会自动作为主服务器进行接管。

`synchronize_mirror_on_commit` 选项用于控制在异步或异步满页模式下运行时保证何时将数据库更改发送到镜像服务器。此选项设置为 `On` 时，每个 COMMIT 都会导致将事务日志中记录的所有更改发送到镜像服务器，并致使镜像服务器在收到更改后向主服务器发送确认。可以使用 `SET TEMPORARY OPTION` 为特定事务设置此选项。通过在登录过程中检查 `APPINFO` 字符串为特定应用程序设置此选项也很有用。

SQL Anywhere 支持当数据库镜像系统中发生故障转移时所触发的那些系统事件，而不管您使用的是哪种模式。您可将这些事件用于诸如发生故障转移时向管理员发出通知之类的任务。请参见“[数据库镜像系统事件](#)”一节第 901 页。

另请参见

- [“synchronize_mirror_on_commit 选项 \[数据库\]”一节第 551 页](#)
- [“-xp 数据库选项”一节第 244 页](#)
- [“SET OPTION 语句”一节《SQL Anywhere 服务器 - SQL 参考》](#)

同步状态

镜像系统使用同步模式时，可处于以下两种状态之一：正在同步或已同步。

当运行服务器启动并确定其将担当镜像服务器后，它会首先从主服务器请求其尚未包含的任何日志页。除了从主服务器上的当前活动日志复制日志页外，这还包括从日志文件复制日志页。镜像服务器收到这些日志页后，会将其中包含的更改应用于它的数据库副本。在收到来自主服务器的所有日志页后，主服务器和镜像服务器便处于已同步状态。自此，主服务器上提交的任何更改必须发送到镜像服务器并由镜像服务器加以确认。

在异步和异步满页模式下，镜像服务器也会如上所述来请求日志页；但这两个服务器永远也不会进入已同步状态。在镜像服务器请求了主服务器上可用的所有日志页后，主服务器会得到通知，必须向镜像服务器发送任何已更新的页面。

状态信息文件

镜像系统中的每个服务器都维护有一个状态信息文件，其中记录了在该服务器的角度上镜像系统的状态。

在启动期间确定服务器要担当的角色时要用到该状态信息文件。将服务器的本地状态与数据库镜像系统中其它服务器的状态进行比较。必须使用 `-xf` 选项为镜像系统中的每个服务器始终指定一个状态信息文件。请参见 [“-xf 服务器选项”一节第 223 页](#)。

状态信息文件包含以下信息：

字段	说明
Owner	指明哪个数据库服务器是主服务器。
State	包含用以指明服务器是正在接收日志页还是处于最新状态的同步状态（正在同步或已同步）。请参见 “同步状态”一节第 885 页 。
Mode	指定同步模式（同步、异步或页面）。请参见 “选择数据库镜像模式”一节第 883 页 。
Sequence	包含一个值，指明数据库镜像系统中已发生的故障转移次数。每切换一次角色，该序列号就会递增 1。它可帮助确定在服务器的角度上，镜像系统的状态是否为最新。请参见 “数据库镜像简介”一节第 880 页 。

下面显示了状态信息文件的内容示例：

```
[demo]
Owner=server2
```

```
State=synchronizing  
Mode=asynchronous  
Sequence=35
```

如果状态信息文件不存在，则自动创建。状态信息文件只能由数据库服务器进行修改。

教程：使用数据库镜像

本教程介绍如何设置数据库镜像系统以及在发生故障转移时会出现什么情况。就本教程而言，所有数据库服务器均运行在同一台计算机上。但在实际镜像系统中，数据库服务器很可能运行在不同的计算机上。

◆ 模拟数据库镜像系统中的故障转移

1. 创建以下目录：`c:\server1`、`c:\server2` 和 `c:\arbiter`。
2. 为 `samples-dir\demo.db` 中的示例数据库制作一个副本，然后将该副本添加到 `c:\server1`。
有关 `samples-dir` 的信息，请参见“[示例目录](#)”一节第 368 页。
3. 通过执行以下命令，为 `c:\server1` 中的数据库创建一个事务日志：

```
dbping -d -c "UID=DBA;PWD=sql;DBF=c:\server1\demo.db"
```

4. 为 `c:\server1` 中的数据库文件和事务日志制作副本，然后将这些副本添加到 `c:\server2`。
5. 运行以下命令启动仲裁服务器：

```
dbsrv11 -x tcpip(PORT=2639) -su sql -n arbiter -xa "auth=abc;DBN=demo" -xf c:\arbiter\arbiterstate.txt
```

此命令行指定以下 `mlsrv11` 选项：

- **-x** 指示数据库服务器使用 TCP/IP，通过端口 2639 进行通信。其它服务器也使用 TCP/IP，但通过不同的端口进行通信。
 - **-su** 指定实用程序数据库的口令。
 - **-n** 将数据库服务器命名为 `arbiter`。
 - **-xa** 指定正在镜像的数据库的名称，并为仲裁服务器指定验证字符串（本例中为 `abc`）。必须将此验证字符串用于数据库镜像系统中的所有服务器（仲裁服务器、主服务器和镜像服务器）。
 - **-xf** 指定 `arbiter` 的状态信息文件的位置。
6. 运行以下命令（此命令必须在一行中键入）启动 `server1`：

```
dbsrv11 -n server1 -x tcpip(PORT=2638) -xf c:\server1\server1state.txt -su sql c:\server1\demo.db -sn mirrordemo -xp "partner=(ENG=server2;LINKS=tcpip(PORT=2637;TIMEOUT=1));auth=abc;arbiter=(ENG=arbiter;LINKS=tcpip(PORT=2639;TIMEOUT=1));mode=sync"
```

此命令行指定以下 `mlsrv11` 选项：

- **-n** 将数据库服务器命名为 `server1`。
- **-x** 指定数据库服务器运行时使用的端口。
- **-xf** 指定 `server1` 的状态信息文件的位置。
- **-su** 指定实用程序数据库的口令。

- **-sn** 指定数据库服务器的替代名称。主服务器和镜像服务器必须具有相同的名称，这样客户端在连接时就无需预先知道哪个服务器是主服务器，哪个服务器是镜像服务器。
- **-xp** 向正在启动的服务器提供信息，以便它能够连接到其合作伙伴和仲裁服务器。

7. 运行以下命令（此命令必须在一行中键入）启动 `server2`：

```
dbsrv11 -n server2 -x tcpip(PORT=2637) -xf c:\server2\server2state.txt
-su sql c:\server2\demo.db -sn mirrordemo
-xp "partner=(ENG=server1;LINKS=tcpip(PORT=2638;TIMEOUT=1));auth=abc;
arbitrator=(ENG=arbitrator;LINKS=tcpip(PORT=2639;TIMEOUT=1));mode=sync"
```

此命令行指定以下 `dbsrv11` 选项：

- **-n** 将数据库服务器命名为 `server2`。
- **-x** 指定数据库服务器运行时使用的端口。
- **-xf** 指定 `server2` 的状态信息文件的位置。
- **-su** 指定实用程序数据库的口令。
- **-sn** 指定数据库服务器的替代名称。主服务器和镜像服务器必须具有相同的名称，这样客户端在连接时就无需预先知道哪个服务器是主服务器，哪个服务器是镜像服务器。
- **-xp** 向正在启动的服务器提供信息，以便它能够连接到其合作伙伴和仲裁服务器。

8. 通过运行以下命令启动 `Interactive SQL` 并连接到主服务器：

```
dbisql -c "UID=DBA;PWD=sql;ENG=mirrordemo;LINKS=tcpip"
```

9. 通过执行以下语句将示例数据添加到 `SQL Anywhere` 示例数据库：

```
CREATE TABLE test (col1 INTEGER, col2 CHAR(32));
INSERT INTO test VALUES(1, 'Hello from server1');
COMMIT;
```

10. 通过执行以下语句确定连接到的是哪个数据库服务器：

```
SELECT PROPERTY( 'ServerName' );
```

将出现主服务器的名称。

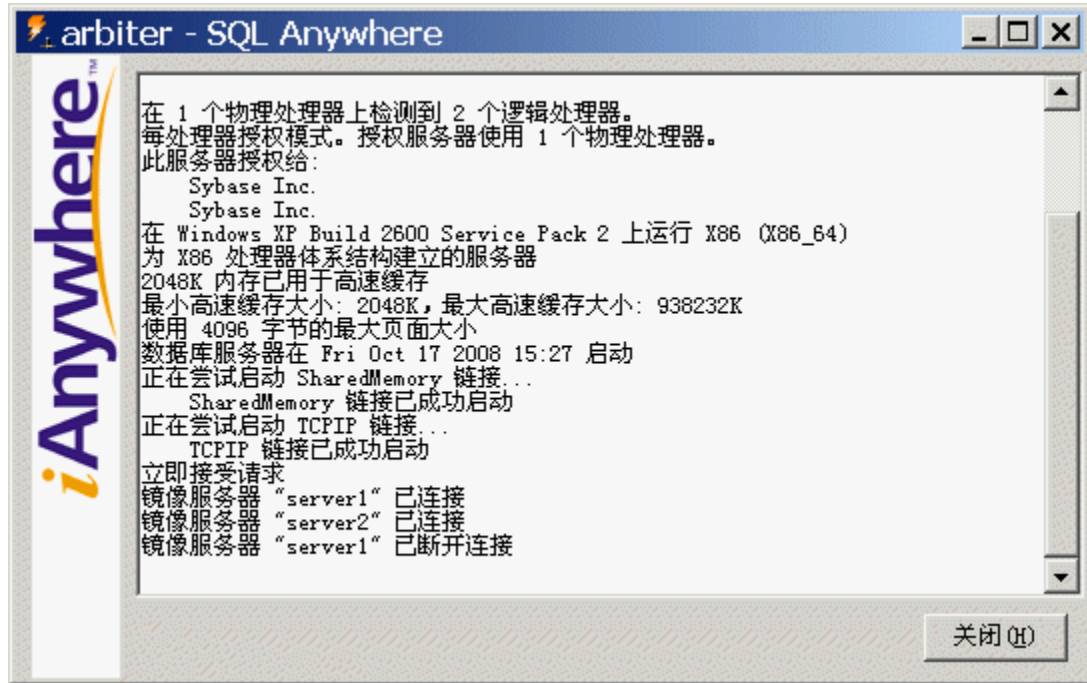
11. 启动故障转移。要实现此目的，通过以下方式之一停止在上一步骤中所识别的主服务器：

- 在数据库服务器消息窗口中单击 **[关闭]**。
- 使用 **[Windows 任务管理器]** 结束其任务。
- 发出以下命令：

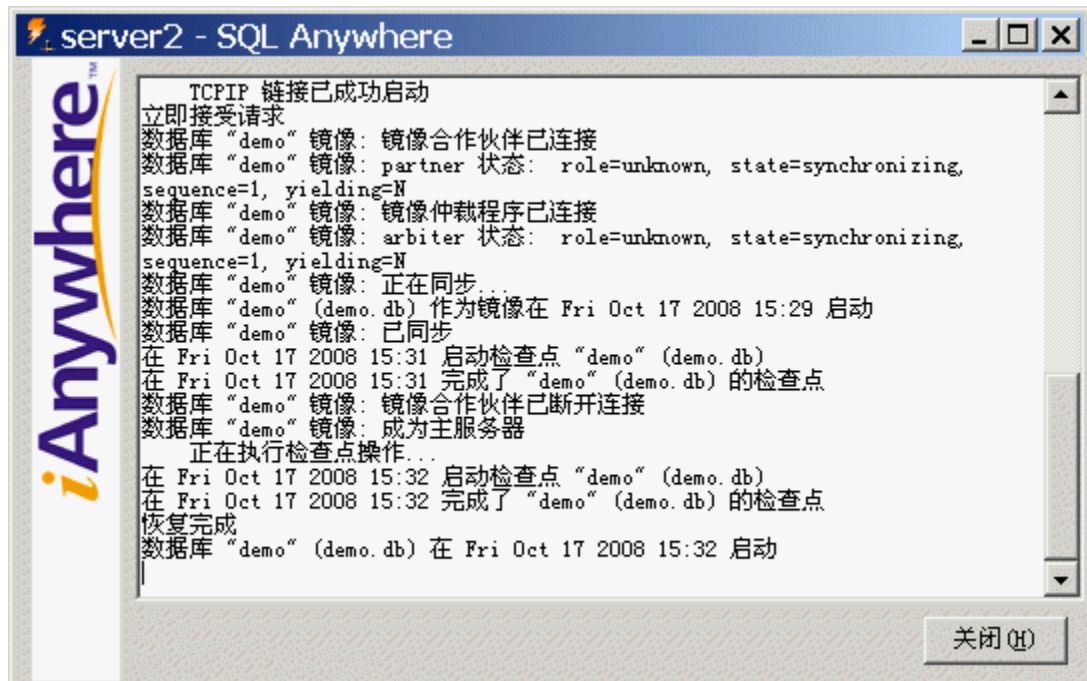
```
dbstop -y -c "UID=DBA;PWD=sql;ENG=mirrordemo"
```

如果出现警告消息，指出该数据库服务器仍存在一个连接，请单击 **[是]** 将其关闭。

仲裁数据库服务器消息窗口将显示一条消息，指出主服务器已断开连接。



server2 数据库服务器消息窗口将显示一条消息，指出该服务器是新的主服务器：



12. 关闭 Interactive SQL。如果您收到错误消息，请单击 [确定]。

13. 通过运行以下命令重新启动 Interactive SQL:

```
dbisql -c "UID=DBA;PWD=sql;ENG=mirrordemo;LINKS=tcPIP"
```

14. 执行以下语句以验证您当前是否已连接到镜像服务器:

```
SELECT PROPERTY ( 'ServerName' );
```

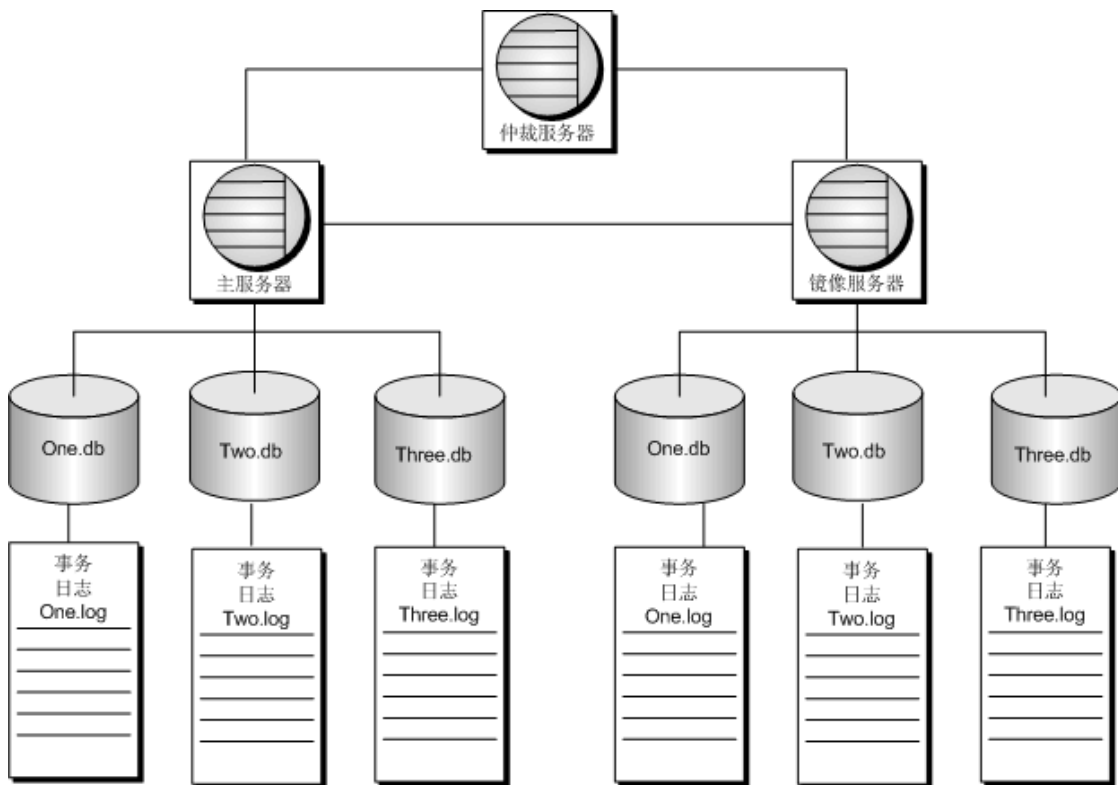
15. 执行以下语句以验证是否已将所有事务均镜像到镜像服务器:

```
SELECT * FROM test;
```

16. 与 Interactive SQL 断开连接, 然后分别在 `arbiter` 和 `server2` 数据库服务器的数据库服务器消息窗口中单击 [关闭]。

教程：使用多个数据库共享一个仲裁服务器的数据库镜像

在此配置中，主服务器和镜像服务器每个服务器上都有参与镜像系统的三个单独的数据库。所有三个镜像系统都与同一个仲裁服务器通信。每个镜像系统均使用使用 `-sn` 选项指定的唯一的替代服务器名。使用这种类型的配置，主服务器、镜像服务器和仲裁服务器全部可以在单独的计算机上运行。



如果主服务器变为不可用，则出现角色切换，镜像服务器取得数据库所有权。镜像服务器变为主服务器。客户端必须重新建立与主服务器的连接。重新建立与主服务器的连接时只需指定替代服务器名。此配置还能够防止单个数据库出现故障。如果在主服务器上运行的数据库变为不可用，则出现角色切换，镜像服务器取得出现故障的数据库的所有权。镜像服务器只变为此数据库的主服务器。客户端必须使用替代服务器名重新建立与此数据库的主服务器的连接。

◆ 设置三个数据库和一个仲裁服务器的镜像系统

1. 创建以下目录：

- `c:\server1`
- `c:\server2`
- `c:\arbiter`

2. 从 `c:\server1` 目录运行以下命令：

```
dbinit one.db
```

```
dbinit two.db
```

```
dbinit three.db
```

3. 通过运行以下命令，为每个数据库创建一个事务日志：

```
dbping -d -c "UID=DBA;PWD=sql;DBF=c:\server1\one.db"
```

```
dbping -d -c "UID=DBA;PWD=sql;DBF=c:\server1\two.db"
```

```
dbping -d -c "UID=DBA;PWD=sql;DBF=c:\server1\three.db"
```

4. 将数据库从 `c:\server1` 目录复制到 `c:\server2` 目录。
5. 启动仲裁服务器：

```
dbsrv11  
-x tcpip(port=2640)  
-n arbiter  
-xa "AUTH=abc,def,ghi;DBN=one,two,three"  
-xf c:\arbiter\arbiterstate.txt  
-su sql
```

6. 启动 `server1` 上的数据库：

```
dbsrv11  
-n server1  
-x tcpip(PORT=2638)  
-xf c:\server1\server1state.txt  
-su sql  
c:\server1\one.db  
-sn mirrortutorial_one  
-xp "partner=(ENG=server2;LINKS=tcpip(PORT=2639;TIMEOUT=1));  
auth=abc;arbiter=(ENG=arbiter;LINKS=tcpip(PORT=2640;TIMEOUT=1));  
mode=sync"  
c:\server1\two.db  
-sn mirrortutorial_two  
-xp "partner=(ENG=server2;LINKS=tcpip(PORT=2639;TIMEOUT=1));  
auth=def;arbiter=(ENG=arbiter;LINKS=tcpip(PORT=2640;TIMEOUT=1));  
mode=sync"  
c:\server1\three.db  
-sn mirrortutorial_three  
-xp "partner=(ENG=server2;LINKS=tcpip(PORT=2639;TIMEOUT=1));  
auth=ghi;arbiter=(ENG=arbiter;LINKS=tcpip(PORT=2640;TIMEOUT=1));  
mode=sync"
```

7. 启动 `server2` 上的数据库：

```
dbsrv11  
-n server2  
-x tcpip(PORT=2639)  
-xf c:\server2\server2state.txt  
-su sql  
c:\server2\one.db  
-sn mirrortutorial_one  
-xp "partner=(ENG=server1;LINKS=tcpip(PORT=2638;TIMEOUT=1));  
auth=abc;arbiter=(ENG=arbiter;LINKS=tcpip(PORT=2640;TIMEOUT=1));  
mode=sync"  
c:\server2\two.db  
-sn mirrortutorial_two  
-xp "partner=(ENG=server1;LINKS=tcpip(PORT=2638;TIMEOUT=1));  
auth=def;arbiter=(ENG=arbiter;LINKS=tcpip(PORT=2640;TIMEOUT=1));  
mode=sync"  
c:\server2\three.db
```

```
-sn mirrortutorial_three
-xp "partner=(ENG=server1;LINKS=tcpip(PORT=2638;TIMEOUT=1));
auth=ghi;arbiter=(ENG=arbiter;LINKS=tcpip(PORT=2640;TIMEOUT=1));
mode=sync"
```

启动 server2 后，server1 数据库服务器消息窗口显示 server1 是数据库一、二和三的镜像系统的主服务器。这些消息还指出一、二和三（合作伙伴）的镜像数据库连接到 server1。

仲裁消息显示 server1 和 server2 均已连接。

8. 运行以下命令启动 Interactive SQL 并连接到主服务器上的数据库一：

```
dbisql -c "UID=DBA;PWD=sql;ENG=mirrortutorial_one;LINKS=TCPIP"
```

9. 通过执行以下语句将示例数据添加到 SQL Anywhere 示例数据库：

```
CREATE TABLE test (col1 INTEGER, col2 CHAR(32));
INSERT INTO test VALUES(1, 'Hello from server1');
COMMIT;
```

10. 通过执行以下语句确定连接的是哪个数据库服务器：

```
SELECT PROPERTY( 'ServerName' );
```

将出现主服务器的名称。

11. 与 Interactive SQL 断开连接。

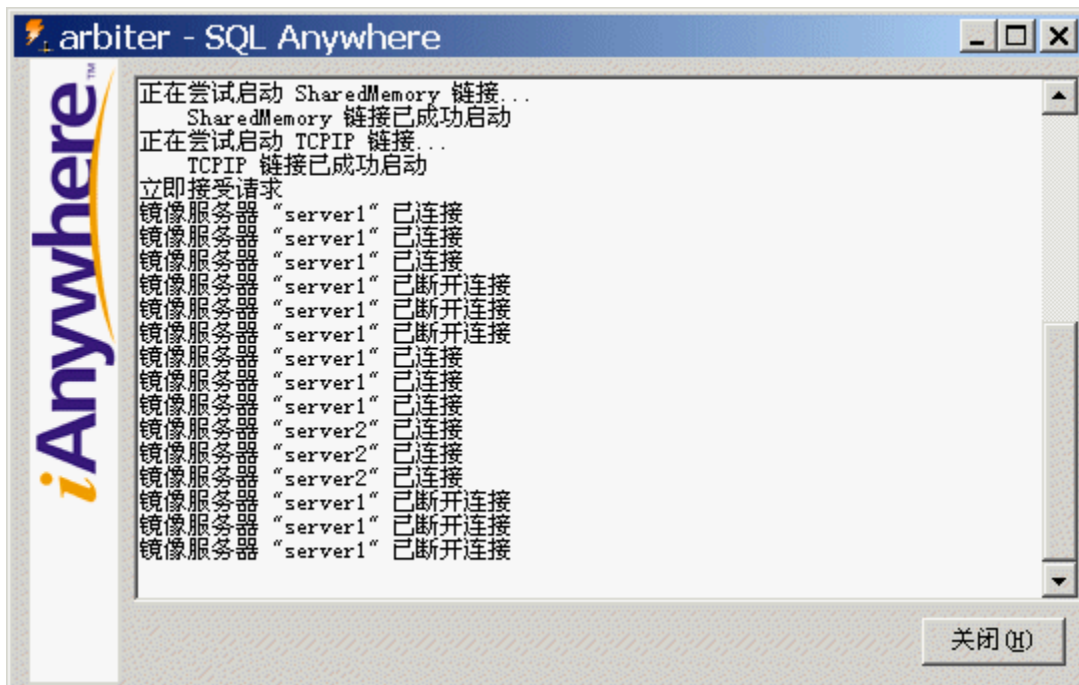
12. 启动故障转移。要实现此目的，可以通过以下方式之一停止主服务器：

- 在数据库服务器消息窗口中单击 **[关闭]**。
- 使用 **[Windows 任务管理器]** 结束其任务。
- 发出以下命令：

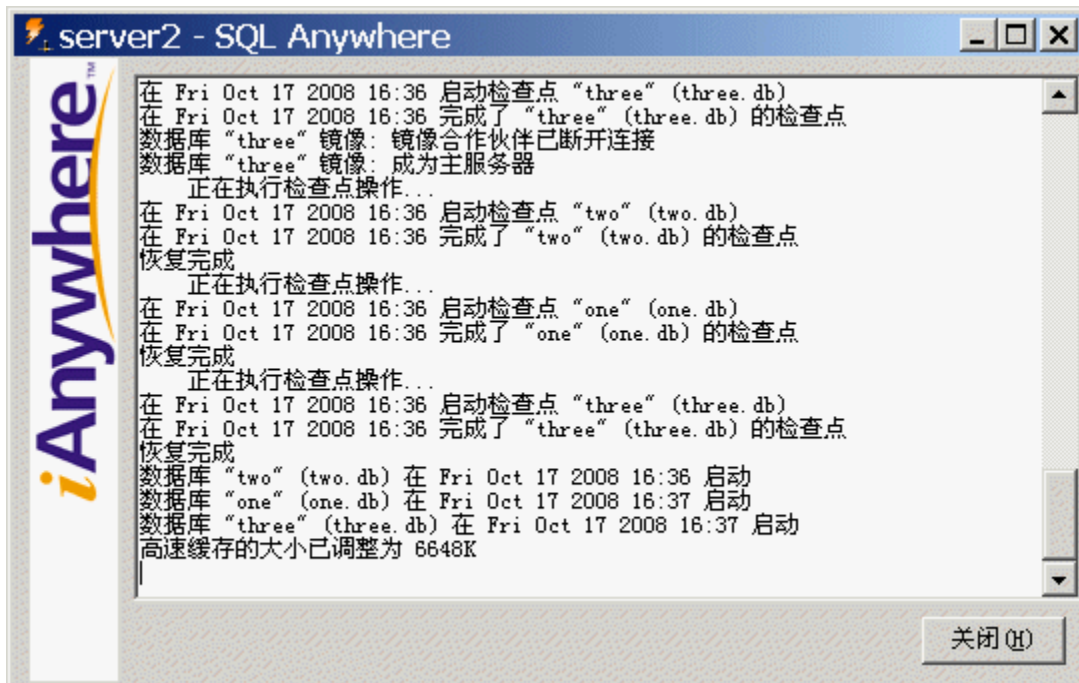
```
dbstop -y -c "UID=DBA;PWD=sql;ENG=server1"
```

如果出现警告消息，指出该数据库服务器仍存在一个连接，请单击 **[是]** 将其关闭。

仲裁数据库服务器消息窗口将显示一条消息，指出主服务器已断开连接。



server2 数据库服务器消息窗口将显示一条消息，指出该服务器是新的主服务器：



13. 通过运行以下命令重新启动 Interactive SQL:


```
dbisql -c "UID=DBA;PWD=sql;ENG=mirrortutorial_one;LINKS=tcPIP"
```

14. 执行以下语句以验证您当前是否已连接到镜像服务器：

```
SELECT PROPERTY ( 'ServerName' );
```

15. 执行以下语句以验证是否已将所有事务均镜像到镜像服务器：

```
SELECT * FROM test;
```

16. 与 Interactive SQL 断开连接，然后分别在 arbiter 和 server2 数据库服务器的数据库服务器消息窗口中单击 **[关闭]**。

设置数据库镜像

以下步骤假定一个数据库服务器已经在运行您想要为其设置镜像系统的数据库。

在启动即将参与镜像系统的数据库服务器时，建议您加入 `-su` 选项以指定实用程序数据库的口令。这样，您可以使用实用程序数据库来关闭服务器，或者强制镜像服务器成为主服务器（在需要这样做的情况下）。请参见“[-su 服务器选项](#)”一节第 211 页。

有关在数据库镜像系统中涉及的升级 SQL Anywhere 或重建数据库的信息，请参见“[在数据库镜像系统中升级 SQL Anywhere 软件和数据库](#)”一节《SQL Anywhere 11 - 更改和升级》。

◆ 设置镜像系统

1. 制作数据库和当前事务日志的副本，将其保存在另外一个服务器上。

如果现有数据库服务器已停止，您可以复制文件；否则使用 `BACKUP DATABASE` 语句或备份实用程序 (`dbbackup`)。请参见“[BACKUP 语句](#)”一节《SQL Anywhere 服务器 - SQL 参考》和“[备份实用程序 \(dbbackup\)](#)”一节第 698 页。

2. 停止正在运行的数据库服务器，修改其命令行配置使其包括镜像选项，然后启动该服务器。

例如：

```
dbsrv11 -n server1 -x tcpip(PORT=2638) -xf c:\server1\server1state.txt
-su sql c:\server1\mirrordemo.db -sn mirrordemo
-xp "partner=(ENG=server2;LINKS=tcpip(PORT=2637;TIMEOUT=1));auth=abc;
arbiter=(ENG=arbiter;LINKS=tcpip(PORT=2639;TIMEOUT=1));mode=page;autofailo
ver=YES"
```

3. 启动另一个运行服务器。

例如：

```
dbsrv11 -n server2 -x tcpip(port=2637) -xf c:\server2\server1state.txt
-su sql c:\server2\mirrordemo.db -sn mirrordemo
-xp "partner=(ENG=server1;LINKS=tcpip(PORT=2638;TIMEOUT=1));auth=abc;
arbiter=(ENG=arbiter;LINKS=tcpip(PORT=2639;TIMEOUT=1));mode=page;autofailo
ver=YES"
```

4. 启动仲裁服务器。

例如：

```
dbsrv11 -x tcpip -n arbiter
-xa "AUTH=abc;DBN=mirrordemo" -xf arbiterstate.txt
-su sql
```

客户端现在可以连接到镜像数据库。

连接到镜像数据库服务器

连接到镜像数据库时，客户端必须使用在用于启动主服务器和镜像服务器的命令中由 `-sn` 选项指定的服务器名。使用上面的示例时（数据库服务器是使用 `-sn mirrordemo` 选项启动的），客户端可在各自的连接字符串中指定连接参数 `ENG=mirrordemo`：

```
...UID=user12;PWD=x92H4pY;ENG=mirrordemo;LINKS=tcpip...
```

如果主服务器和镜像服务器在不同的子网中运行，还必须指定客户端连接到主服务器时应使用的 IP 地址范围。例如：

```
...UID=user12;PWD=x92H4pY;ENG=mirrordemo;LINKS=tcpip(HOST=ip1,ip2...)...
```

您可能还需要指定 `RetryConnectionTimeout` 连接参数，以控制客户端可以在多长时间内重试与主服务器之间的连接尝试。请参见“[RetryConnectionTimeout 连接参数 \[RetryConnTO\]](#)”一节第 279 页。

如果不能顺利找到客户端需要连接的服务器，请尝试执行以下操作：

1. 指定运行主服务器和镜像服务器的计算机的主机名。例如，如果它们分别在名为 `MirrorServ1` 和 `MirrorServ2` 的计算机上运行，则可以在客户端连接字符串中使用 `LINKS=tcpip(HOST=MirrorServ1,MirrorServ2)`。
2. 使用 LDAP 注册服务器。请参见“[使用 LDAP 服务器进行连接](#)”一节第 138 页。
3. 使用 SQL Anywhere 广播转发器实用程序 (`dbns11`) 定位服务器。此实用程序用于监听一个子网上的广播和响应，然后在另一个子网上再次广播。请参见“[广播转发器实用程序 \(dbns11\)](#)”一节第 703 页。

确定初始主服务器

第一次设置数据库镜像系统时，没有状态信息文件，数据库和事务日志的副本完全相同，两个服务器均有资格担当主服务器。在这种情况下，比较服务器名，名称小的服务器担当主服务器。例如，名称 `server1` 比 `server2` 小。

初始启动时，两个服务器都必须运行并连接，以对角色达成一致；只有仲裁服务器是不够的，因为在状态信息文件中记录的先前状态信息不存在。

正常启动时，以下输入影响哪个服务器变为主服务器：

- 状态信息文件的内容
- 每个数据库服务器上事务日志位置
- 首选主服务器的指定

另请参见

- “[状态信息文件](#)”一节第 885 页

指定首选数据库服务器

在数据库镜像系统中，可以将两个运行服务器中的一个标识为首选服务器。如果所有数据库服务器都在运行，则首选服务器将成为主服务器，并取得数据库的所有权。如果标记为首选的服务器不可用，则充当镜像服务器的服务器将成为主服务器。当首选服务器重新启动时，它会获得尚未从当前主服务器得到的全部事务日志条目。接下来，它要求当前主服务器放弃数据库所有权。然后服务器将改变角色，首选服务器将成为主服务器，其它服务器会成为镜像服务器。当数据库所有权更改时，到非首选服务器上数据库的全部连接都将丢失。

启动数据库服务器时，将 "preferred=YES" 添加到 -xp 数据库选项可指定首选服务器。例如：

```
dbsrv11 -n server1 mydata.db -sn mydata
-xp "partner=(ENG=server2;LINKS=tcpip(TIMEOUT=1));
AUTH=abc;arbiter=(ENG=arbsrv;LINKS=tcpip(TIMEOUT=1));preferred=YES"
```

另请参见

- “-xp 数据库选项” 一节第 244 页
- “启动主服务器上的故障转移” 一节第 899 页
- “选择数据库镜像模式” 一节第 883 页

配置在镜像服务器上运行的数据库的只读访问

使用数据库镜像时，可以使用只读连接访问在镜像服务器上运行的数据库。如果您要下放报告或需要此数据库的只读访问的其它操作，此功能很有用。

在镜像系统中，您不必知道哪个数据库担当主服务器和镜像服务器。如果您要能够连接到在镜像服务器上运行的数据库，需在启动数据库服务器时包括 -sm 服务器选项。这样，通过提供用于访问只读镜像数据库的服务器名，就能够建立找到镜像服务器的连接。只有在数据库服务器担当数据库的镜像服务器时，通过 -sm 选项指定的服务器名才处于活动状态。通常对这两个数据库服务器都指定 -sm 选项，因为您不知道哪个服务器担当主服务器或镜像服务器。例如，-sm 选项指示数据库服务器使用 mysamplemirror 作为连接到在镜像服务器上运行的数据库时的替代服务器名：

```
dbsrv11 -n myserver satest.db sample.db -sn mysampleprimary
-sm mysamplemirror
-xp "partner=( ENG=server2;LINKS=TCPIP( PORT=2637;TIMEOUT=1 ) );auth=abc;
arbiter=( ENG=arbiter;LINKS=TCPIP( PORT=2639;TIMEOUT=1 ) );mode=sync"
```

尝试更改数据库将导致错误，其行为与使用 -r 选项作为只读启动数据库时的行为相同。可以在临时表上执行操作，但在镜像数据库上不触发事件。事件触发只在发生从主服务器到镜像服务器的故障转移之后启动。如果已定义，此时触发 DatabaseStart 和 MirrorFailover 事件。有关详细信息，请参见“了解系统事件” 一节第 868 页。

如果发生故障转移，镜像服务器变为主服务器，将维持与镜像服务器的连接。故障转移后，连接可对数据库进行更改。您可以查询 ReadOnly 数据库属性的值，以确定您所连接的数据库是否可以更新：

```
SELECT DB_PROPERTY( 'ReadOnly' );
```

另请参见

- “-sm 数据库选项” 一节第 241 页
- ReadOnly 属性：“数据库属性” 一节第 601 页

针对镜像数据库运行查询

根据指定的隔离级别，针对镜像数据库执行的查询可设置锁。如果锁干扰从主服务器应用的操作，则将回退持有锁的连接的事务，并关闭这些连接的任何打开的游标。以隔离级别 0 运行的应用程序不添加行锁，但仍获取模式锁。如果模式锁干扰从主服务器应用的操作，则将回退针对镜像数据库的事务。

需要一致的数据库视图（所以不能使用隔离级别 0）的应用程序应考虑使用快照隔离。为此，`allow_snapshot_isolation` 选项必须设置为 On。此选项在主服务器和镜像服务器上均生效，所以需要考虑与快照隔离有关的成本。

与镜像服务器的连接受针对主服务器的事务影响，因为这些操作随后通过镜像服务器处理和应用。在主服务器上提交更新的时间与在镜像服务器上更新可用的时间之间有短暂延迟。此延迟通常很短，但访问在镜像服务器上运行的数据库时应记住此延迟。

另请参见

- “`-sm` 数据库选项”一节第 241 页
- “快照隔离”一节 《SQL Anywhere 服务器 - SQL 的用法》
- “`allow_snapshot_isolation` 选项 [数据库]”一节第 478 页

强制数据库服务器成为主服务器

在需要强制主服务器关闭的情况下（例如，如果要更换运行主服务器的计算机），可以使用 `ALTER DATABASE` 语句在镜像服务器通过其它方式不能取得数据库的所有权时强制它成为主服务器。

要使用此功能，必须连接到镜像数据库服务器上的实用程序数据库。通过在启动镜像服务器的命令中指定 `-su` 选项，可以连接到实用程序数据库。以下命令会强制 `mymirroredb.db` 数据库的镜像服务器成为主服务器：

```
ALTER DATABASE mymirroredb FORCE START;
```

`FORCE START` 子句强制当前作为镜像服务器的数据库服务器取得数据库的所有权。此语句可从过程或事件内部执行，并且必须在连接到镜像服务器上的实用程序数据库时执行。请参见“[连接到实用程序数据库](#)”一节第 29 页。

如果要强制从主服务器到镜像服务器的故障转移，可以采取下列措施：

- 停止主服务器
- 在连接到主服务器上的数据库时执行 `ALTER DATABASE SET PARTNER FAILOVER`（此语句会使主服务器重新启动数据库并成为镜像服务器）

另请参见

- “`ALTER DATABASE` 语句”一节 《SQL Anywhere 服务器 - SQL 参考》

启动主服务器上的故障转移

执行以下语句，可以启动从主服务器到镜像服务器的数据库镜像故障转移：

```
ALTER DATABASE SET PARTNER FAILOVER;
```

此语句是指定首选服务器的替代方法，可以结合控制何时将数据库所有权转让给特定数据库服务器的逻辑使用。例如，您可能需要基于伙伴服务器的可用性（由 `PartnerState` 数据库属性的值确定），或者数据库连接的数量（由 `ConnCount` 数据库属性的值确定）来启动故障转移。

执行此语句时，到数据库的全部现有连接都将关闭，包括执行此语句的连接。如果此语句包含在过程或事件中，它后面的其它语句将不会执行。执行此语句的所需权限由 `-gk` 服务器选项控制。

另请参见

- “指定首选数据库服务器”一节第 897 页
- “ALTER DATABASE 语句”一节 《SQL Anywhere 服务器 - SQL 参考》
- “-gk 服务器选项”一节第 181 页
- ConnCount 和 PartnerState 属性：“数据库属性”一节第 601 页

停止镜像系统中的数据库服务器

在有些情况下，您可能需要停止主服务器、镜像服务器或仲裁服务器。为此，您可以使用停止数据库实用程序 (`dbstop`)。

要停止服务器，必须使用与实用程序数据库的连接，因此建议您在启动数据库服务器时加入 `-su` 服务器选项。请参见“使用实用程序数据库”一节第 28 页。

要使用运行镜像数据库的数据库服务器的替代服务器名，启动数据库服务器时必须使用 `-sm` 选项。请参见“-sm 数据库选项”一节第 241 页。

◆ 停止主服务器、镜像服务器或仲裁服务器

- 发出 `dbstop` 命令停止数据库服务器。

例如，以下命令用于停止名为 `myarbiter` 的数据库服务器：

```
dbstop -c "UID=DBA;PWD=sql;DBN=utility_db;LINKS=tcPIP" myarbiter
```

从主服务器故障恢复

从主服务器故障恢复的步骤取决于数据库镜像系统所采用的同步模式。

如果在同步模式下运行，则也会保证主服务器上存在的所有事务都提交到镜像服务器上。镜像服务器可作为新主服务器进行接管，而无需任何用户干预。

在异步或异步满页模式下，从主服务器到镜像服务器的故障转移不会自动进行，因为镜像服务器中可能没有主服务器上所应用的全部已提交事务。在使用异步或异步满页模式的情况下，除非已经指定应该进行自动故障转移，否则缺省情况下，镜像服务器不能在主服务器发生故障时取得数据库的所有权。当重新启动发生故障的服务器时，它会检测是否丢失了事务。如果丢失了事务，则将一条消息写入数据库服务器消息日志，然后关闭数据库。

当启动原镜像服务器作为新的主服务器时，您可以通过以下两种方法将两个服务器上的数据库文件置为同一状态：

- 将原主服务器中的数据库和事务日志文件复制到镜像服务器，然后启动镜像服务器作为新的主服务器。可使用 ALTER DATABASE 语句强制服务器成为主服务器。请参见“ALTER DATABASE 语句”一节 《SQL Anywhere 服务器 - SQL 参考》。

- 在原镜像服务器上执行备份（使用 `dbbackup`）。将文件复制到原主服务器，然后启动这两个数据库服务器。

数据库镜像和事务日志文件

运行服务器启动时，它会检查所有事务日志文件是否都在当前事务日志文件所在的目录中，并确定需要应用哪些文件。随后，数据库服务器会将这些事务日志中的操作应用于数据库，然后确定是担当主服务器还是镜像服务器。

如果服务器担当镜像服务器角色，它会开始接收主服务器中的事务日志页。如果在主服务器上发生事务日志重命名，则重命名同样会在镜像服务器上执行。镜像服务器随后会将新的事务日志页写入具有为事务日志指定的名称的新文件中。

可定期删除主服务器上的事务日志文件。每次重命名事务日志文件时，都会将哪个事务日志文件是主服务器上仍然保留的最早文件通知给镜像服务器。镜像服务器上所有早于该文件的事务日志文件都将被删除。

由于针对请求事务日志截断的主服务器执行备份时镜像服务器可能会不可用，所以必须使用与截断事务日志不同的方法删除主服务器上的事务日志（如使用 `xp_cmdshell` 删除存留时间超过一周的文件的预定事件）。

数据库镜像系统事件

数据库镜像支持以下系统事件：

- **MirrorFailover** 每当数据库服务器取得镜像数据库的所有权时便会触发此事件。例如，在服务器初次启动并确定其将拥有数据库时，便会触发该事件。当先前担当镜像服务器的服务器确定主服务器已出现故障，并在与仲裁服务器协商后确定其应拥有所有权时，也会触发该事件。
- **MirrorServerDisconnect** 当主服务器与镜像服务器或仲裁服务器之间失去连接时，将会触发 `MirrorServerDisconnect` 事件。在此事件的处理程序中，`EVENT_PARAMETER('MirrorServerName')` 的值是失去连接的服务器的名称。

这两个事件不会在当前正在充当镜像服务器的服务器上触发。同样，不能将镜像事件定义为在仲裁服务器上执行，因为事件仅在定义它们的数据库环境中运行，并且仲裁服务器也不会使用要被镜像的数据库副本。

您可将这两个事件用作通知机制，以电子邮件的形式将可能需要对镜像数据库采取行动的通知发送出去。在所有会导致在主服务器上运行的数据库变为不可用的情况下都不会触发这两个事件。例如，同时影响主服务器和镜像服务器的断电情况便不会触发这两个事件中的任何一个事件。如果需要这种类型的监控，可以通过脚本编写语言在单独的计算机上加以实现，方法是通过调用 `dbping` 定期连接到镜像服务器。请参见“[Ping 实用程序 \(dbping\)](#)”一节第 757 页。

以下示例创建了一个事件，用以在发生故障转移时通知管理员：

```
CREATE EVENT mirror_server_unavailable
TYPE MirrorServerDisconnect
HANDLER
BEGIN
    CALL xp_startmail ( mail_user ='George Smith',
```

```
                mail_password = 'mypwd' );  
CALL xp_sendmail( recipient='DBAdmin',  
                subject='Database failover occurred',  
                "message"='The following server is unavailable in the mirroring system: '  
                || event_parameter( 'MirrorServerName' ) );  
CALL xp_stopmail ( );  
END;
```

另请参见

- “了解系统事件”一节第 868 页

数据库镜像和性能

理想情况下，应该为运行主服务器和镜像服务器的计算机配置相似的硬件（处理器、磁盘、存储器等）。在任何给定时间，在任何一台计算机上运行的数据库服务器都可以充当要被镜像数据库的主服务器。镜像服务器的利用率通常很低，这取决于主服务器上的更新活动。

镜像并不影响针对主服务器的查询性能。更新数据库的事务的性能取决于事务的大小和提交的频率。在异步模式下运行的镜像服务器要比在同步模式下运行的镜像服务器具有更佳的性能，但速度仍旧比没有参与镜像系统的数据库服务器的速度慢。性能高低在很大程度上取决于运行服务器之间的网络连接速度。

数据库镜像和备份

虽然数据库镜像会有助于将数据丢失的风险降到最低，但仍建议对参与数据库镜像系统的数据库进行备份和校验。

可使用 BACKUP DATABASE 语句来执行与数据库服务器相关的备份。BACKUP DATABASE 语句是在主数据库服务器上执行的，所以提供的文件名应指定一个对于主数据库服务器和镜像数据库服务器而言一致的网络驱动器或 UNC 名称。请参见“BACKUP 语句”一节《SQL Anywhere 服务器 - SQL 参考》。

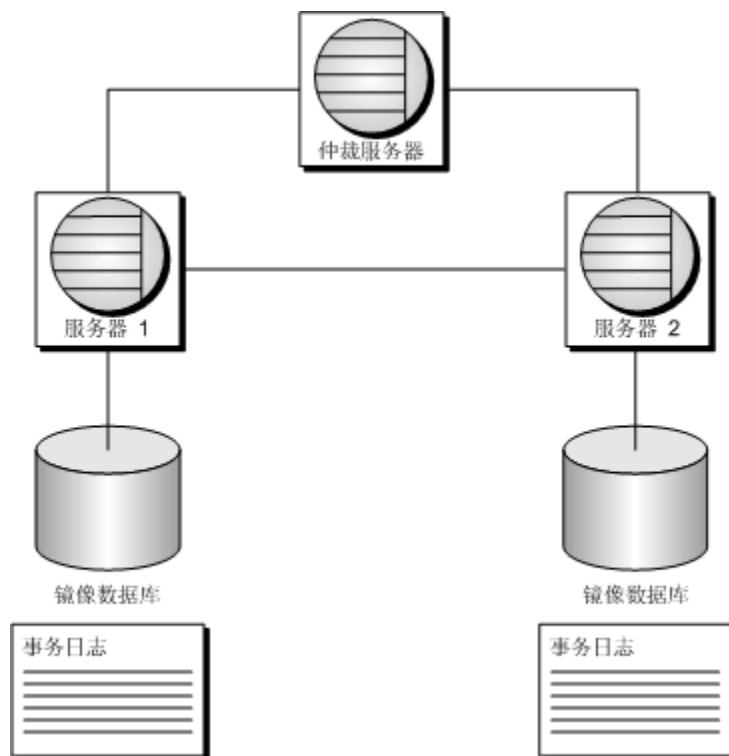
另外，还可使用 dbbackup 实用程序来执行客户端备份。请参见“备份实用程序 (dbbackup)”一节第 698 页。

另请参见

- “备份和数据恢复”第 817 页
- “校验简介”一节第 858 页

数据库镜像情形

以下情形会帮助您了解当镜像系统中的服务器变为不可用时会发生什么情况。这些情形使用以下数据库镜像配置，其中包括在同步模式下运行的服务器 1、服务器 2 和仲裁服务器：



您可以随时使用 MirrorState、PartnerState 和 ArbiterState 数据库属性来确定镜像系统中数据库服务器的状态。请参见“[数据库属性](#)”一节第 601 页。

第 1 种情形：主服务器变为不可用

1. 主服务器（服务器 1）变为不可用。所有客户端断开连接。
2. 仲裁服务器和服务器 2 检测到服务器 1 不再可用。
3. 仲裁服务器和服务器 2 达到法定数目，服务器 2 成为主服务器。
4. 服务器 2 开始接受客户端连接。

在这种情形下，如果在异步或异步满页模式下运行并且没有指定应该发生自动故障转移，则可能需要制作数据库的副本并重新启动仍在运行的服务器，之后客户端才能重新连接。

有关在主服务器变为不可用的情况下进行恢复的详细信息，请参见“[从主服务器故障恢复](#)”一节第 900 页。

第 2 种情形：主服务器变为不可用后重新启动

1. 仲裁服务器和镜像服务器（服务器 2）检测到主服务器（服务器 1）不再可用。
2. 仲裁服务器和服务器 2 达到法定数目，服务器 2 成为主服务器。
3. 服务器 2 开始接受客户端连接。
4. 服务器 1 恢复在线并重新连接到服务器 2 和仲裁服务器。

5. 服务器 1 请求法定数目，但服务器 2 已经是主服务器。
6. 服务器 1 是镜像服务器，等待来自服务器 2 的更改。
7. 服务器 2 向服务器 1 发送更改。

如果在服务器 1 还没有收到服务器 2 的所有事务的情况下，服务器 2 已变为不可用，那么服务器 1 将不能达到已同步状态。它必须等待服务器 2 重新变为可用，才能够获得和应用目前还没拥有的事务。

有关在主服务器变为不可用的情况下进行恢复的详细信息，请参见“[从主服务器故障恢复](#)”一节第 900 页。

第 3 种情形：镜像服务器变为不可用

1. 镜像服务器（服务器 2）变为不可用。
2. 仲裁服务器和服务器 1 检测到镜像服务器（服务器 2）不再可用。

客户端连接不受影响。它们可以继续连接到主服务器。但是，如果服务器 1 或仲裁服务器变为不可用，客户端便无法连接。

第 4 种情形：镜像服务器变为不可用后重新启动

1. 镜像服务器（服务器 2）变为不可用。
2. 客户端连接不受影响，因为可用性方面没有变化。它们可以继续连接到主服务器。但是，如果服务器 1 或仲裁服务器变为不可用，客户端便无法连接。
3. 服务器 2 恢复在线并重新连接到服务器 1 和仲裁服务器。
4. 服务器 2 请求法定数目，但服务器 1 已经是主服务器。
5. 服务器 2 是镜像服务器，等待来自服务器 1 的更改。
6. 服务器 1 向服务器 2 发送更改。

客户端连接不受影响，因为可用性方面没有变化。它们继续连接到服务器 1。

第 5 种情形：仲裁服务器变为不可用

1. 服务器 1（主服务器）和服务器 2（镜像服务器）检测到仲裁服务器不可用。
2. 而这两个服务器仍然可用。客户端不会断开连接。

仲裁服务器恢复在线后，服务器 1 和服务器 2 会检测到它并开始与其通信。对于客户端来说，数据库可用性方面没有变化。

如果在没有仲裁服务器的情况下，服务器 1 或服务器 2 变为不可用，则另一个服务器自己无法达到法定数目，同时数据库将不再可用。

第 6 种情形：仲裁服务器重新启动

1. 仲裁服务器恢复在线并重新连接到服务器 1 和服务器 2。

客户端连接不受影响，因为可用性方面没有变化。

使用 SQL Anywhere Veritas 聚簇服务器代理

需要单独授予许可的组成部分

SQL Anywhere Veritas 聚簇服务器代理需要单独的许可。请参见“单独授权的组件”一节《SQL Anywhere 11 - 简介》。

聚簇是称为**节点**的一组计算机，它们协同工作以运行一系列应用程序。与运行在某聚簇上的应用程序相连接的客户端会将该聚簇视为单个系统。如果某个节点发生故障，聚簇中的其它节点能够自动接管该故障节点所提供的服务。在可用性方面，客户端可能会察觉到微小的中断（恢复其余节点上的服务所需的时间），要不然不会知道该节点发生了故障。

聚簇与 SQL Anywhere 结合使用时，数据库或数据库服务器故障转移到聚簇中其它节点时会丢失所有尚未提交的事务，同时，在发生故障转移后，客户端必须重新连接到数据库。

SQL Anywhere 支持多种聚簇环境，在这些环境中，聚簇软件可将任何应用程序转变为通用资源，以便能够进行自动故障转移，从而提供高可用性。但是，只有数据库服务器进程能够进行故障转移，监控和控制进程不能进行故障转移。

有关详细信息，请参见 <http://www.sybase.com/detail?id=1034743>。

大多数聚簇软件都为创建适合于特定应用程序的自定义资源提供了 API。SQL Anywhere 包括两个适合于 Veritas 聚簇服务器的自定义故障转移资源：SAServer 和 SADBatabase。SAServer 代理负责数据库服务器故障转移，SADBatabase 代理负责特定数据库文件的故障转移。您可使用其中一个代理，也可同时使用两个代理，这要视应用程序而定。

要使用 SQL Anywhere Veritas 聚簇服务器代理，系统必须进行如下设置：

- 必须使用 Veritas Cluster Server 4.1 或更高版本。
- 在聚簇中的每个系统节点上必须以完全相同的方式来安装 SQL Anywhere。
- 数据库文件必须存储在聚簇中所有系统均可访问的共享存储设备上。
- 聚簇中所有系统的实用程序数据库口令必须相同。

SADBatabase 代理使用实用程序数据库启动和停止特定数据库文件。参与聚簇的所有系统必须具有相同的实用程序数据库口令。通过在启动数据库服务器时指定 `-su` 服务器选项，可以设置实用程序数据库口令。

在 Unix 上，VCS 代理安装在 `install-dir/vcsagent/saserver` 中。

有以下三种方法用于将新代理配置和添加到 Veritas 聚簇服务器：

1. 使用聚簇管理器。
2. 使用命令行实用程序。
3. 使用文本编辑器并编辑 `main.cf` 配置文件。

以下各节中的说明使用的是聚簇管理器。

有关可用实用程序的信息，请参见 *Veritas Cluster Server Administration Guide*。

如果要使用文本编辑器手工配置 *main.cf*，则在编辑 *main.cf* 文件之前必须停止所有 Veritas 聚簇服务器服务。否则，更改将不生效。

配置 SAServer 代理

SAServer 代理用于控制 SQL Anywhere 数据库服务器到聚簇中其它节点的故障转移。

◆ 设置 SAServer 代理

1. 关闭正在聚簇中各节点上运行的所有 SQL Anywhere 数据库服务器。
2. 在聚簇中选择一个节点，在此节点上的 *%VCS_HOME%\bin* 目录下创建名为 *SAServer* 的目录。此文件夹中还会列出其它 Veritas 聚簇服务器代理（如 *NIC* 和 *IP*）。
3. 将以下文件从 *install-dir\VCSAgent\SAServer* 目录复制到在第 2 步中创建的 *SAServer* 目录：
 - *Online.pl*
 - *Offline.pl*
 - *Monitor.pl*
 - *Clean.pl*
 - *SAServer.xml*
4. 将文件 *%VCS_HOME%\bin\VCSdefault.dll* 复制到 *%VCS_HOME%\bin\SAServer* 目录，并将其重命名为 *SAServer.dll*。
5. 将文件 *install-dir\VCSAgent\SAServer\SAServerTypes.cf* 复制到 *%VCS_HOME%\conf\config* 目录。
6. 对于聚簇中的所有其它节点，重复执行第 1 步到第 5 步。
7. 启动 Veritas Cluster Server Manager，并输入您的用户名和口令以连接到聚簇。
8. 添加 SAServer 代理：
 - a. 选择 [**File**] » [**Import Types**]。
 - b. 浏览到 *%VCS_HOME%\conf\config\SAServerTypes.cf*，然后单击 [**Import**]。

◆ 使用 SAServer 代理设置数据库服务器进行故障转移

1. 启动 Veritas 聚簇服务器管理器，并输入您的用户名和口令进行连接。
2. 向服务组中添加 SAServer 作为一种资源：
 - a. 选择 [**Edit**] » [**Add**] » [**Resource**]。
 - b. 在 [**Resource Type**] 列表中，选择 [**SAServer**]。

在 Windows 上，如果 SAServer 未出现在 Windows 下的 [**Resource Type**] 列表中，则需要将 *SAServer.xml* 文件添加到 *%VCS_ROOT%\cluster manager\attrpool\Win2K\400*，然后重新启动聚簇服务。
 - c. 在 [**Resource Name**] 字段中键入名称。
 - d. 将以下属性值添加到以下属性：

- **cmdStart** dbsrv11 -x tcpip *database-file-on-shared-disk* -n *server-name*
 - **cmdMonitor** dbping -c "ENG=*server-name*"
 - **cmdStop** dbstop -c *user-id,password* -y
- e. 选择 **[Enabled]**。
这表示该资源可供使用。
 - f. 单击 **[OK]**。
3. 确保正确配置资源依赖性。在启动 SAServer 之前还必须启动和组合其它资源，如共享磁盘资源和 IP 地址资源。
 4. 右击服务组，然后选择 **[Online]** » [*node-name*]，其中 *node-name* 是聚簇中要运行该资源的计算机的名称。
现在，服务组在线。

测试 SAServer 代理

以下步骤介绍如何测试 SAServer 代理的故障转移情况。

◆ 测试 SAServer 代理故障转移

1. 从 Interactive SQL 连接到数据库。例如：

```
dbisql -c "UID=DBA;PWD=sql;ENG=VCS;LINKS=tcpip"
```

2. 执行以下查询：

```
SELECT * FROM Departments;
```

该查询应无错误地执行。

3. 关闭运行数据库服务器的系统。
应发生故障转移，并且应在备用服务器上启动所有资源。
4. 从 Interactive SQL 使用同一个连接字符串重新连接，然后再次执行该查询。您应该能够成功地连接和执行查询。

配置 SADatabase 代理

SADatabase 代理用于控制 SQL Anywhere 数据库到聚簇中其它节点的故障转移。

◆ 设置 SADatabase 代理

1. 关闭正在聚簇中各节点上运行的所有 SQL Anywhere 数据库服务器。
2. 在聚簇中的一个节点上创建名为 `%VCS_HOME%\bin\SADatabase` 的目录。

3. 将以下文件从 *install-dir\SADatabase* 目录复制到在第 2 步中创建的 *%VCS_HOME%\bin\SADatabase* 目录：
 - *Online.pl*
 - *Offline.pl*
 - *Monitor.pl*
 - *Clean.pl*
 - *SADatabase.xml*
4. 将文件 *%VCS_HOME%\bin\VCSdefault.dll* 复制到 *%VCS_HOME%\bin\SADatabase* 目录，并将其重命名为 *SADatabase.dll*。
5. 将文件 *install-dir\SADatabase\SADatabaseTypes.cf* 复制到 *%VCS_HOME%\conf\config* 目录。
6. 对于参与聚簇的所有系统，重复执行第 1 步到第 5 步。
7. 启动 Veritas 聚簇服务器管理器，并输入您的用户名和口令以连接到聚簇。
8. 添加 SADatabase 代理：
 - a. 从 [File] 菜单中选择 [Import Types]。
 - b. 浏览到 *%VCS_HOME%\conf\config*，然后单击 [Import]。

◆ 使用 SADatabase 代理设置数据库进行故障转移

1. 向服务组中添加 SADatabase 作为一种资源：
 - a. 从 [Edit] 菜单选择 [Add] » [Resource]。
 - b. 在 [Resource Type] 列表中，选择 [SADatabase]。

在 Windows 上，如果 SADatabase 未出现在 [Resource Type] 列表中，则需要将 *SADatabase.xml* 文件添加到 *%VCS_ROOT%\cluster manager\attrpool\Win2K\400*，然后重新启动聚簇服务。
 - c. 在 [Resource Name] 字段中键入名称。
 - d. 通过单击每个属性的 [Edit] 列中的按钮，将指定的值添加到以下属性中：
 - **DatabaseFile** 数据库文件的位置，例如 *E:\demo.db*。
 - **DatabaseName** 数据库的名称。
 - **ServerName** 数据库服务器的名称。可为聚簇中的每个系统提供不同的服务器名称。该属性的范围应为 Per System，而非 Global。
 - **UtilDBpwd** 用于聚簇中所有系统的实用程序数据库口令。
 - e. 选择 [Enabled]。

这表示该资源可供使用。
 - f. 单击 [OK]。
2. 确保正确配置资源依赖性。在启动 SADatabase 之前还要必须启动/组合其它资源，如共享磁盘资源和 IP 地址资源。
3. 右击服务组，然后选择 [Online] » [node-name]，其中 *node-name* 是聚簇中要运行该资源的计算机的名称。

现在，服务组在线。

测试 SADatabase 代理

以下步骤介绍如何测试 SADatabase 代理的故障转移情况。

◆ 测试 SADatabase 代理故障转移

1. 从 Interactive SQL 连接到数据库。例如：

```
dbisql -c "UID=DBA;PWD=sql;ENG=VCS;LINKS=tcPIP"
```

2. 执行以下查询：

```
SELECT * FROM Departments;
```

该查询应无错误地执行。

3. 假定数据库发生故障，并且在第一个系统节点上运行的数据库服务器不能访问数据库文件。这将创建数据库文件到第二个系统节点上启动的数据库服务器的故障转移。通过发出以下类似命令会导致第一个节点上的数据库文件发生故障：

```
dbisql -q -c "UID=DBA;PWD=sql;ENG=VCS1;DBN=utility_db" STOP DATABASE DEMO  
ON VCS1 UNCONDITIONALLY;
```

第一台计算机上的数据库文件发生故障。在 Veritas 聚簇服务器能够识别出该文件已出现故障之前会有一个延迟，因为在缺省情况下，Veritas 聚簇服务器会每隔 60 秒钟监控一次其资源的运行状况（可在资源配置中将此间隔设置小些）。随后，该数据库文件会故障转移到第二台计算机上，并使用第二台计算机上的数据库服务器启动该数据库文件，这个数据库服务器的名称可能与原数据库服务器的名称不同。

例如，如果新数据库服务器称为 VCS2，则客户端必须在各自的连接字符串中指定新的数据库服务器名称：

```
"UID=DBA;PWD=sql;ENG=VCS2;DBN=DEMO;LINKS=tcPIP"
```

4. 从 Interactive SQL 重新连接。您应该能够成功地连接和执行查询。

监控数据库

本节介绍如何使用 SQL Anywhere 监控器来监控您的 SQL Anywhere 数据库和 MobiLink 服务器。还将介绍如何设置和配置 SQL Anywhere SNMP Extension Agent。

SQL Anywhere 监控器	913
SQL Anywhere SNMP Extension Agent	955

SQL Anywhere 监控器

目录

SQL Anywhere 监控器简介	914
监控器快速入门	917
教程：使用监控器	918
启动监控器	923
退出监控器	924
连接到监控器	925
断开与监控器的连接	926
监控资源	927
管理资源	935
使用监控器用户	942
警告	946
安装的对象	950
在一台单独的计算机上安装 SQL Anywhere 监控器	951
监控器疑难解答	952

SQL Anywhere 监控器简介

SQL Anywhere 监控器（也称为监控器）是一种基于 Web 浏览器的管理工具，可提供有关 SQL Anywhere 数据库和 MobiLink 服务器健康情况和可用性的信息。

本章介绍如何使用监控器来收集有关 SQL Anywhere 数据库的度量。有关将监控器与 MobiLink 服务器配合使用的信息，请参见“用于 MobiLink 的 SQL Anywhere 监控器”《MobiLink - 服务器管理》。

此监控器提供以下功能：

- **常量数据收集** 与 SQL Anywhere 11 提供的许多其它管理工具不同，监控器会一直收集度量，甚至在您未登录到 Web 浏览器时也是如此。除非将此监控器关闭，否则其会一直收集度量。
- **电子邮件警告通知** 当收集到度量时，此监控器会对度量进行检查，并在检测到指示数据库出现错误的条件时发送电子邮件警告。
- **基于浏览器的界面** 您可以随时使用 Web 浏览器连接到此监控器，以查看警告和收集到的度量。
- **监控多个数据库和 MobiLink 服务器** 通过同一个工具，您可以同时监控在一台计算机或不同计算机上运行的 SQL Anywhere 数据库和 MobiLink 服务器。
有关监控 MobiLink 服务器的信息，请参见“用于 MobiLink 的 SQL Anywhere 监控器”《MobiLink - 服务器管理》。
- **最小化性能影响** 因为监控不会降低性能，所以可以在开发与生产环境中例行使用此监控器。

要求

- 建议安装可用于操作系统的最新版 Adobe Flash Player。监控器向后兼容 Adobe Flash Player 版本 9。要确定正确的版本，请访问 <http://www.adobe.com/products/flashplayer/systemreqs/>。
- 必须在 Web 浏览器中启用 JavaScript。
- 必须已安装 SQL Anywhere 11.0.1。

在生产环境中运行监控器

可以在一台单独的计算机上安装并运行监控器。这样可以防止监控器资源和配置在后续的 SQL Anywhere 升级或更新过程中被覆盖。如果要在生产环境中使用监控器，建议在一台单独的计算机上安装监控器。请参见“在一台单独的计算机上安装 SQL Anywhere 监控器”一节第 951 页。

限制

- 可使用监控器收集有关以下类型 SQL Anywhere 数据库和 MobiLink 服务器的度量：
 - SQL Anywhere 9.0.2、10.0.0、10.0.1、11.0.0 和 11.0.1
 - MobiLink 11.0.0（至少应用了第一个 EBF）和 11.0.1
- 在一台计算机上只能运行一个监控器。

- 不能使用此监控器优化查询或确定应用程序的速度。如果您对调整数据库和应用程序性能感兴趣，则可以使用 [应用程序分析向导]、Sybase Central 性能监控器或 Windows 性能监控器之类的工具。

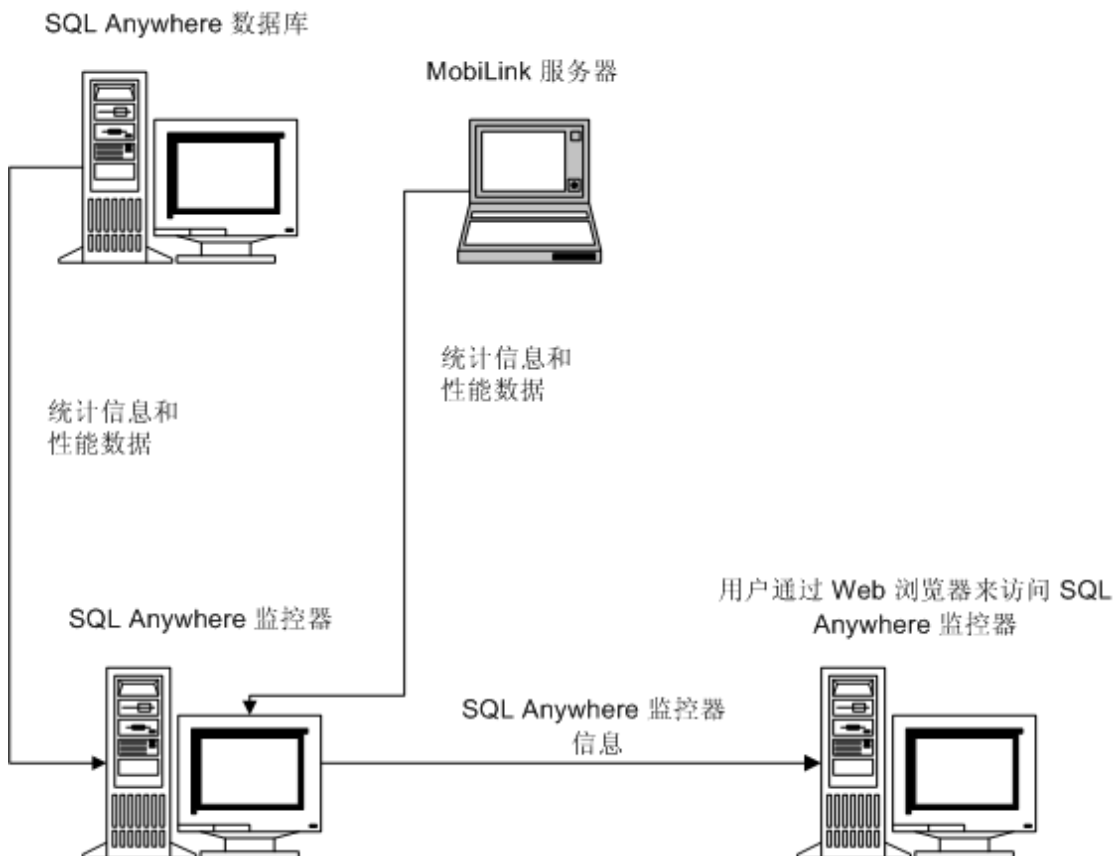
另请参见

有关可用于 SQL Anywhere 数据库的其它管理工具和性能工具的信息，请参见：

- “应用程序分析”一节 《SQL Anywhere 服务器 - SQL 的用法》
- “SQL Anywhere 控制台实用程序 (dbconsole)”一节第 780 页
- “使用 Sybase Central 性能监控器监控统计信息”一节 《SQL Anywhere 服务器 - SQL 的用法》
- “使用 Windows 性能监控器监控统计信息”一节 《SQL Anywhere 服务器 - SQL 的用法》

监控器体系结构

此监控器从运行于其它计算机上的 SQL Anywhere 数据库和 MobiLink 服务器收集度和性能数据，而单独的计算机通过 Web 浏览器来访问此监控器。



此监控器设计用于帮助任何类型的用户（无论他们是否为 DBA），这些用户负责如下任务：

- 确保数据库已连接到网络。
- 确保数据库有足够的磁盘空间或内存可用。
- 确保不会阻挡用户或者查询不会占用太长时间。

另请参见

- [“监控器快速入门”一节第 917 页](#)

监控器快速入门

设置 SQL Anywhere 数据库监控需要执行以下步骤：

1. 将 SQL Anywhere 11.0.1 安装到始终与网络相连的一台计算机上。此监控器使用 SQL Anywhere 来监控数据库。
监控器可以与其所监控的资源运行在同一台计算机上，但建议您在另一台计算机上运行监控器（在生产环境中时尤其如此），以将对数据库服务器或其它应用程序的影响降至最低。
2. 确保您的 Web 浏览器已安装了适当版本的 Adobe Flash Player 并且已启用了 JavaScript。请参见“[要求](#)”一节第 914 页。
3. 启动数据库（如果其尚未运行）。
4. 启动监控器并在 Web 浏览器中将其打开。请参见“[启动监控器](#)”一节第 923 页。
在其中使用 Web 浏览器来访问监控器的计算机必须连接到此监控器运行的网络。
5. 以管理员身份登录。缺省用户名为 **admin**，缺省口令也是 **admin**。
6. 单击 **[管理]** 选项卡，并添加 SQL Anywhere 数据库作为要监控的资源。请参见“[添加资源](#)”一节第 935 页。
7. 添加新用户并更改管理员用户口令。请参见“[创建监控器用户](#)”一节第 942 页。
8. 为要监控的数据库配置警告。请参见“[警告](#)”一节第 946 页。
9. 单击 **[监控]** 选项卡可查看为数据库收集的度量。请参见“[监控资源](#)”一节第 927 页。

教程：使用监控器

使用本教程设置对 SQL Anywhere 示例数据库的监控。

第 1 课：启动监控器

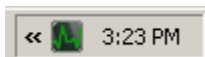
◆ 启动并打开监控器

1. 启动监控器。选择 [开始] » [程序] » [SQL Anywhere 11] » [SQL Anywhere 监控器] » [启动 SQL Anywhere 监控器]。

如果是在一台单独的计算机上安装监控器，则不必执行此步骤。如果安装监控器的计算机不同于 SQL Anywhere 所运行的计算机，则监控器将作为服务来运行并在计算机启动时自动启动。

2. 浏览数据。根据监控器是否在一台单独的计算机上安装，此步骤会有所不同。

在系统任务栏中，右击 SQL Anywhere 监控器图标并选择 [浏览数据]。



如果监控器安装在一台单独的计算机上，选择 [开始] » [程序] » [SQL Anywhere 监控器 11] » [浏览数据]。系统任务栏中不会出现图标。

或者，您也可以打开 Web 浏览器，然后浏览到 <http://localhost:4950>。

资源	状态	监控器状态	启动时间	上次检查
SQL Anywhere Monitor	活动	良好	2009/05/26 17:20	2009/05/26 17:38

时间	状态	警告

[监控] 选项卡的顶部窗格列出了正在受监控的资源。首次打开监控器时，其只监控它本身。

另请参见

- “第 2 课：设置监控器来监控数据库” 一节第 919 页

第 2 课：设置监控器来监控数据库

此监控器从数据库和 MobiLink 服务器收集度量。在本节中，您将启动 SQL Anywhere 示例数据库 *demo.db*，然后将此数据库添加为要监控的资源。要从 MobiLink 服务器收集度量，请参见“第 2 课：设置监控器来监控 MobiLink 服务器”一节《MobiLink - 服务器管理》。

◆ 将资源添加到监控器

1. 启动 SQL Anywhere 示例数据库。从 [开始] 菜单选择 [程序] >> [SQL Anywhere 11] >> [SQL Anywhere] >> [网络服务器示例]。
2. 以缺省管理员身份登录到监控器：
 - a. 单击 [登录]。
 - b. 在 [用户名] 字段中，键入 **admin**，然后在 [口令] 字段中键入 **admin**。
 - c. 单击 [登录]。
3. 单击 [管理] 选项卡。
4. 单击 [资源] 选项卡。
5. 单击 [添加]。
6. 选择 [SQL Anywhere 服务器]，然后单击 [下一步]。
7. 将资源命名为 **demo11**，然后单击 [下一步]。
8. 在 [主机] 字段中，键入 **localhost**，然后在 [服务器] 字段中键入 **demo11**。
9. 单击 [创建]。
10. 当系统提示您需要授权时，在 [DBA 用户 ID] 字段中键入 **DBA**，然后在 [口令] 字段中键入 **sql**。单击 [确定]。

监控器会将监控对象安装在 **demo11** 数据库中。将创建新资源 **demo11**，并且将开始监控。

11. 单击 [确定]。
12. 单击 [监控] 选项卡。

demo11 资源将出现在 [监控] 选项卡上，收集到的度量出现在底部窗格的选项卡上。

第 3 课：测试警告

在本课中，您将有意地触发警告，以便能够练习如何处理警告。

◆ 查看和解析警告

1. 通过关闭 **demo11** 数据库触发警告。

- a. 在 Windows 上，双击系统任务栏中 demo11 数据库服务器的网络服务器图标。
 - b. 在数据库服务器消息窗口中单击 [关闭]。
 - c. 单击 [是]。
2. 在监控器中，单击 [监控] 选项卡。
- demo11 资源的 [状态] (State) 将变为 [数据库关闭]；demo11 资源的 [状态] (Status) 将变为 [需要注意!]。
- 状态 (State) 和状态 (Status) 可能在几秒钟之后才会发生变化。缺省情况下，监控器将每隔 30 秒从资源收集一次信息。
3. 在底部窗格中，单击 [警告]。
 4. 选择 [可用性警告]，然后单击 [详细信息] 以读取说明。
 5. 单击 [确定]。
 6. 重新启动示例数据库。
- 启动 SQL Anywhere 示例数据库。从 [开始] 菜单选择 [程序] ? [SQL Anywhere 11] ? [SQL Anywhere] ? [网络服务器示例]。
- Demo11 资源的 [状态 (State)] 将变为 [活动]，而 [状态 (Status)] 不会发生变化。变化要在几分钟之后才会出现。
7. 通过选择警告并单击 [删除] 来将其删除。
- [状态] 将变为 [良好]。

第 4 课：设置监控器以在出现警告时发送电子邮件

当警告出现时，其总是会列在 [监控] 选项卡的下部窗格中的 [警告] 选项卡内。在以下步骤中，将设置监控器以在出现警告时向您发送电子邮件。

◆ 设置电子邮件通知

1. 创建一个可接收电子邮件的用户。
 - a. 单击 [管理] 选项卡。
 - b. 单击 [用户] 选项卡。
 - c. 单击 [新建]。
 - d. 在 [用户名] 字段中键入 JoeSmith。
 - e. 在 [口令] 和 [确认口令] 字段中键入 sql。
 - f. 在 [电子邮件] 字段中输入有效的电子邮件地址。
 - g. 在 [首选语言] 字段中，选择 [中文]。
 - h. 为 [用户类型] 选择 [操作员]。

操作员可通过电子邮件接收警告，并可解析和删除警告。该用户可访问 **[监控]** 选项卡，但是不能访问 **[管理]** 选项卡。

有关不同类型用户的信息，请参见“[使用监控器用户](#)”一节第 942 页。

- i. 单击 **[保存]**。
即会创建新用户。
2. 将用户与 **demo11** 资源相关联。
 - a. 单击 **[资源]** 选项卡。
 - b. 选择 **[demo11]** 资源，然后单击 **[配置]**。
 - c. 在 **[配置资源]** 窗口中，单击 **[操作员]**。
 - d. 在 **[可用操作员]** 列表中，选择 **JoeSmith** 并单击 **[添加]**。
 - e. 单击 **[保存]**。
 - f. 单击 **[确定]**。
3. 配置电子邮件警告通知。
 - a. 单击 **[管理]** 选项卡。
 - b. 单击 **[配置]** 选项卡。
 - c. 单击 **[编辑]**。
 - d. 选择 **[通过电子邮件发送警告通知]**。
 - e. 根据需要配置其它设置。
 - f. 测试您是否已正确配置了电子邮件通知。
单击 **[发送测试电子邮件]**。
 - g. 在系统提示时，输入要将测试电子邮件发送到的电子邮件地址，然后单击 **[确定]**。
测试电子邮件将会被发送到指定的电子邮件地址。
 - h. 单击 **[保存]**。

发生警告时，会将带有关于此警告的信息的电子邮件发送给指定用户。有关设置警告的信息，请参见“[第 3 课：测试警告](#)”一节第 919 页。

第 5 课：清除

以下步骤将删除 **demo11** 资源，进而删除收集到的度量并停止数据收集。在生产环境中，当您想要持续监控数据库时，可使数据库和监控器处于运行状态。

◆ 停止监控

1. 删除 **demo11** 资源。
 - a. 单击 **[管理]** 选项卡。
 - b. 单击 **[资源]** 选项卡。

- c. 选择 demo11 资源，然后单击 **[停止]**。
 - d. 单击 **[删除]**。
 - e. 单击 **[是]**，确认您确实要删除该资源。
2. 注销监控器。
单击 **[注销]**。
3. 关闭您正在其中查看监控器的 Web 浏览器窗口。
4. 退出监控器。
在系统任务栏中，右击 SQL Anywhere 监控器图标并选择 **[退出 SQL Anywhere 监控器]**。
5. 关闭 SQL Anywhere 数据库。
 - a. 双击系统任务栏中 demo11 数据库服务器的网络服务器图标。
 - b. 在数据库服务器消息窗口中单击 **[关闭]**。
 - c. 单击 **[是]**。

启动监控器

启动监控器会使监控器开始为此监控器中的*所有*资源收集度量。

根据监控器是否运行在一台单独的计算机上，启动监控器的过程会有所不同。

◆ 启动监控器

1. 选择 [开始] » [程序] » [SQL Anywhere 11] » [SQL Anywhere 监控器] » [启动 SQL Anywhere 监控器]。

SQL Anywhere 监控器图标出现在系统任务栏中。

2. 连接到监控器。请参见“[连接到监控器](#)”一节第 925 页。

◆ 启动安装在单独计算机上的监控器

1. 如果监控器安装在单独的计算机上，则它将作为服务而自动运行。但是，如果停止监控，您可以重新启动它。为此，请到 *install-dir\bin32*。
2. 在 Windows 上，运行以下命令：

```
samonitor.bat start service
```

在 Linux 上，运行以下命令：

```
samonitor.sh start service
```

当监控器作为服务来运行时，系统任务栏中不会出现 SQL Anywhere 监控器图标。

3. 连接到监控器。请参见“[连接到监控器](#)”一节第 925 页。

另请参见

- “[退出监控器](#)”一节第 924 页
- “[连接到监控器](#)”一节第 925 页
- “[断开与监控器的连接](#)”一节第 926 页
- “[监控资源](#)”一节第 927 页

退出监控器

退出监控器将会停止为所有资源收集度量。建议您使监控器处于运行状态，而将 Web 浏览器关闭。要停止监控特定的数据库，请参见“[停止监控资源](#)”一节第 938 页。

根据监控器是否运行在一台单独的计算机上，退出监控器的过程会有所不同。

◆ 退出监控器

- 在系统任务栏中，右击 SQL Anywhere 监控器图标并选择 [[退出 SQL Anywhere 监控器](#)]。

◆ 退出安装在单独计算机上的监控器

1. 到 `install-dir\bin32`。
2. 在 Windows 上，运行以下命令：

```
samonitor.bat stop service
```

在 Linux 上，运行以下命令：

```
samonitor.sh stop service
```

另请参见

- “[启动监控器](#)”一节第 923 页
- “[连接到监控器](#)”一节第 925 页
- “[断开与监控器的连接](#)”一节第 926 页
- “[监控资源](#)”一节第 927 页

连接到监控器

用于连接到监控器的计算机必须连接到此监控器正在运行的网络。

◆ 连接到监控器

1. 启动监控器（如果其尚未运行）。请参见“[启动监控器](#)”一节第 923 页。
2. 浏览数据。根据监控器是否在一台单独的计算机上安装，此步骤会有所不同。

从 [开始] 菜单中，选择 [程序] » [SQL Anywhere 11] » [SQL Anywhere 监控器] » [浏览数据]。

如果监控器安装在一台单独的计算机上，选择 [开始] » [程序] » [SQL Anywhere 监控器 11] » [浏览数据]。

Web 浏览器将打开用于连接到此监控器的缺省 URL: **http://computer-name:4950**，其中 *computer-name* 是正在运行监控器的计算机的名称。例如，*http://localhost:4950*。

3. 如果系统提示，则输入监控器的用户名和口令。监控器的用户名和口令是区分大小写的。请参见“[使用监控器用户](#)”一节第 942 页。

另请参见

- “[启动监控器](#)”一节第 923 页
- “[退出监控器](#)”一节第 924 页
- “[断开与监控器的连接](#)”一节第 926 页
- “[监控资源](#)”一节第 927 页

断开与监控器的连接

可通过注销或关闭 Web 浏览器断开与监控器的连接。

断开与监控器的连接并不会对度量的收集造成影响。如果要停止对度量的收集，则请停止对资源的监控或退出监控器。请参见“[停止监控资源](#)”一节第 938 页或“[退出监控器](#)”一节第 924 页。

◆ 断开与监控器的连接

- 单击 [\[注销\]](#)。

另请参见

- [“启动监控器”一节第 923 页](#)
- [“退出监控器”一节第 924 页](#)
- [“连接到监控器”一节第 925 页](#)
- [“监控资源”一节第 927 页](#)

监控资源

在监控器中，[\[监控\]](#) 选项卡会提供有关所监控 SQL Anywhere 数据库的健康和可用性的概要信息。

[监控] 选项卡

顶部窗格包含一个列有正在受监控的资源的表格。**资源**是指数据库。该表还指明资源当前是否运行以及资源是否需要用户对其执行任何操作。请参见“[解释资源状态 \(states\) 和状态 \(status\)](#)”一节第 928 页。

[\[监控\]](#) 选项卡的底部窗格包含所选数据库的警告和多种当前度量。这些选项卡中的大部分选项卡都包含指向图形的链接。可使用每个图形右上方的下拉列表和箭头更改图形的范围。

[管理] 选项卡

[\[管理\]](#) 选项卡为管理员所保留。在该选项卡上，您可以选择所要监控的数据库，添加和编辑用户，以及配置监控器。



另请参见

- “[使用监控器用户](#)”一节第 942 页
- “[监控器度量](#)”一节第 928 页

解释资源状态 (states) 和状态 (status)

[**监控**] 选项卡的顶部窗格包含一个列出了正在受监控的 SQL Anywhere 数据库的表格。在此表中，[**状态 (State)**] 列提供了关于监控器与其资源之间的连接的信息。[**状态 (Status)**] 列指明该资源是否需要操作员或管理员用户对其执行操作。请参见“[使用监控器用户](#)”一节第 942 页。

资源状态 (state)

资源始终处于以下状态之一：

- **活动** 资源已连接且监控器正在收集度量。
- **封锁** 监控器在其重新开始监控资源之前，正在等待封锁期结束。
- **数据库关闭** 正在受监控的 SQL Anywhere 数据库已停止。
- **主机关闭** 监控器无法找到承载资源的计算机。
- **未知** 此监控器没有监控资源。

资源状态 (status)

资源具有以下状态之一：

- **良好** 资源没有未解决的警告。
- **需要注意** 资源有一个或多个警告。
- **监控已停止** 资源未受监控。
- **未知** 资源未活动且没有警告。

监控器度量

监控器从数据库收集并存储度量，其中包括（但不局限于）：

- 资源是否正在运行。
- 资源正在其中运行的计算机是否运行正常以及是否连接到网络。
- 资源是否正在监听和处理请求。
- 是否存在任何明显的问题，例如长时间运行的查询或受到阻挡的用户。

度量的收集速率是由管理员设置的收集间隔设置所决定的。请参见“[收集间隔](#)”一节第 936 页。

收集哪些度量以及使用什么阈值来发出警告，由管理员设置的度量设置所决定。请参见“[指定要收集的度量](#)”一节第 936 页。

显示度量

监控器的显示将每分钟自动刷新一次。可通过单击 [**用户设置**] 更改刷新间隔。该设置与资源的收集间隔速率无关，收集间隔速率用于指定监控器从所监控资源收集度量的频率。

◆ 设置刷新率

1. 单击右上角的 [用户设置]。
2. 为 [刷新间隔] 设置时间。缺省值为一分钟。
3. 单击 [确定]。

在 [监控] 选项卡上单击 [刷新数据] 时，监控器会检索并显示最新的度量。

◆ 刷新度量

- 单击 [刷新数据]。

按 F5 键时，监控器会重装 Web 浏览器，以及检索和显示监控器到目前为止所收集的度量。

◆ 重装监控器

- 按 F5。

[度量] 选项卡说明

以下选项卡由 SQL Anywhere 和 MobiLink 服务器资源使用。

- “[监控] 选项卡: [警告] 选项卡” 一节第 930 页
- “[监控] 选项卡: [服务器] 选项卡” 一节第 930 页

以下选项卡仅由 SQL Anywhere 资源使用。

- “[监控] 选项卡: [CPU] 选项卡” 一节第 931 页
- “[监控] 选项卡: [未调度的请求] 选项卡” 一节第 931 页
- “[监控] 选项卡: [内存] 选项卡” 一节第 931 页
- “[监控] 选项卡: [磁盘] 选项卡” 一节第 932 页
- “[监控] 选项卡: [HTTP] 选项卡” 一节第 932 页
- “[监控] 选项卡: [连接] 选项卡” 一节第 932 页
- “[监控] 选项卡: [失败的连接] 选项卡” 一节第 933 页
- “[监控] 选项卡: [查询] 选项卡” 一节第 933 页
- “[监控] 选项卡: [镜像] 选项卡” 一节第 933 页

以下选项卡仅由 MobiLink 服务器资源使用。

- “[监控] 选项卡: [同步] 选项卡” 一节 《MobiLink - 服务器管理》
- “[监控] 选项卡: [统一数据库] 选项卡” 一节 《MobiLink - 服务器管理》

- “[监控] 选项卡: [计算机资源] 选项卡”一节 《MobiLink - 服务器管理》

[监控] 选项卡: [警告] 选项卡

列出前五十个最新的警告。一旦列表中的警告数超过 50 个, 则旧的警告即会在新的警告出现时被删除。请参见“警告”一节第 946 页。

[监控] 选项卡: [服务器] 选项卡

SQL Anywhere 服务器

- **服务器名** 当前连接的数据库服务器名称。请参见“数据库服务器属性”一节第 588 页中的 ServerName 属性。
- **数据库名** 显示数据库的名称。请参见“PROPERTY 函数 [System]”一节 《SQL Anywhere 服务器 - SQL 参考》。
- **版本** 显示正在运行的软件版本。请参见“数据库服务器属性”一节第 588 页中的 ProductVersion 属性。
- **类型** 显示正在受监控的数据库服务器的类型。值包括 Personal 和 Network。
- **语言** 显示区域设置语言, 该语言是数据库服务器的用户要使用的语言。请参见“了解区域设置语言”一节第 389 页。
- **开始时间** 显示 SQL Anywhere 数据库启动的时间。
- **未提交的错误报告** 显示数据库的未提交错误报告数。SQL Anywhere 软件崩溃时将提交错误报告。请参见“取消对资源的未提交错误报告的警告”一节第 949 页。

许可

- **类型** 返回许可类型。可以是联网存取设备 (每存取设备) 或基于 CPU 的类型。请参见“数据库服务器属性”一节第 588 页中的 LicenseType 属性。
- **被授权的存取设备数** 显示被授权的存取设备或处理器的数量。请参见“数据库服务器属性”一节第 588 页中的 LicenseCount 属性。
- **被授权公司的名称** 显示被授权公司的名称。请参见“数据库服务器属性”一节第 588 页中的 CompanyName 属性。
- **被授权用户的名称** 显示被授权用户的名称。请参见“数据库服务器属性”一节第 588 页中的 LicensedUser 属性。

Host

- **名称** 显示运行数据库服务器的计算机名称。通常, 这是计算机的主机名。请参见“数据库服务器属性”一节第 588 页中的 MachineName 属性。
- **操作系统平台** 显示运行软件的操作系统。请参见“数据库服务器属性”一节第 588 页中的 Platform 属性。

- **操作系统平台版本** 显示运行软件的操作系统，其中包括内部版本号和服务包。请参见“[数据库服务器属性](#)”一节第 588 页中的 PlatformVer 属性。
- **处理器体系结构** 显示用于标识处理器类型的字符串。请参见“[数据库服务器属性](#)”一节第 588 页中的 ProcessorArchitecture 属性。

另请参见

- [“\[监控\] 选项卡: \[服务器\] 选项卡”](#) 一节第 930 页

[监控] 选项卡: [CPU] 选项卡

监控数据库时使用此选项卡。

- **数据库服务器 CPU 的使用** 显示数据库服务器使用的 CPU 空间百分比。上述百分比基于 ProcessCPU 属性。请参见“[数据库服务器属性](#)”一节第 588 页。

[监控] 选项卡: [未调度的请求] 选项卡

监控数据库时使用此选项卡。

列出当前排队等候可用数据库服务器线程的请求数。请参见“[数据库服务器属性](#)”一节第 588 页中的 UnSchReq 属性。

[监控] 选项卡: [内存] 选项卡

监控数据库时使用此选项卡。

- **当前高速缓存大小** 当前高速缓存大小（以千字节为单位）。请参见“[数据库服务器属性](#)”一节第 588 页中的 CurrentCacheSize 属性。
- **主堆页** 用于全局服务器数据结构的页数。请参见“[数据库服务器属性](#)”一节第 588 页中的 MainHeapPages 属性。
- **高速缓存大小峰值** 高速缓存在当前会话中已达到的最大值（以千字节为单位）。请参见“[数据库服务器属性](#)”一节第 588 页中的 PeakCacheSize 属性。
- **高速缓存已固定** 固定高速缓存页数。请参见“[数据库服务器属性](#)”一节第 588 页中的 CachePinned 属性。
- **高速缓存文件脏** 处于脏状态的高速缓存页数（需要写操作）。请参见“[数据库服务器属性](#)”一节第 588 页中的 CacheFileDirty 属性。
- **高速缓存替换** 高速缓存中已替换的页数。请参见“[数据库服务器属性](#)”一节第 588 页中的 CacheReplacements 属性。

[监控] 选项卡: [磁盘] 选项卡

监控数据库时使用此选项卡。

- **Dbospace:系统** 显示主数据库文件的大小。请参见“[预定义 dbospace](#)”一节第 13 页。
- **Dbospace:事务日志** 显示事务日志的大小。请参见“[预定义 dbospace](#)”一节第 13 页。
- **Dbospace:临时** 显示临时 dbospace 的大小。请参见“[预定义 dbospace](#)”一节第 13 页。
- **磁盘读取** 测量从磁盘读取数据的速率（以每秒千字节为单位）。此值根据 DiskRead 属性进行计算。请参见“[数据库服务器属性](#)”一节第 588 页中的 DiskRead 属性。
- **磁盘写入** 测量数据写入磁盘的速率（以每秒千字节为单位）。此值根据 DiskWrite 属性进行计算。请参见“[数据库服务器属性](#)”一节第 588 页中的 DiskWrite 属性。

[监控] 选项卡: [HTTP] 选项卡

监控数据库时使用此选项卡。

- **会话** 返回数据库服务器中活动的和休眠的 HTTP 会话数量。请参见“[数据库服务器属性](#)”一节第 588 页中的 HttpNumSessions 属性。
- **HTTP 连接** 返回数据库服务器中当前打开的 HTTP 连接数。这些连接可能正在主动处理请求，或者正在长寿命（保持活动状态）连接队列中等待。请参见“[数据库服务器属性](#)”一节第 588 页中的 HttpNumConnections 属性。
- **HTTP 活动请求** 返回正在主动处理 HTTP 请求的 HTTP 连接数。不包括已发送其响应的 HTTP 连接。请参见“[数据库服务器属性](#)”一节第 588 页中的 HttpNumActiveReq 属性。
- **HTTPS 连接** 返回数据库服务器中当前打开的 HTTPS 连接数。这些连接可能正在主动处理请求，或者正在长寿命（保持活动状态）连接队列中等待。请参见“[数据库服务器属性](#)”一节第 588 页中的 HttpsNumConnections 属性。
- **HTTPS 活动请求** 返回正在主动处理 HTTPS 请求的安全 HTTPS 连接数量。不包括已发送其响应的 HTTPS 连接。请参见“[数据库服务器属性](#)”一节第 588 页中的 HttpsNumActiveReq 属性。

[监控] 选项卡: [连接] 选项卡

监控数据库时使用此选项卡。

- **连接计数** 显示数据库的当前连接数。请参见“[sa_conn_info 系统过程](#)”一节《[SQL Anywhere 服务器 - SQL 参考](#)》。
- **存取设备计数** 显示连接到网络数据库服务器的唯一客户端网络地址数。请参见“[数据库服务器属性](#)”一节第 588 页中的 UniqueClientAddresses 属性。

[监控] 选项卡: [失败的连接] 选项卡

监控数据库时使用此选项卡。

列出与数据库之间的失败连接。

[监控] 选项卡: [查询] 选项卡

监控数据库时使用此选项卡。

- **处理的查询** 显示处理查询的速率。请参见“数据库属性”一节第 601 页中的 QueryOptimized、QueryReused 和 QueryBypassed 属性。
- **长时间运行的查询** 列出超出指定的长时间运行查询阈值的查询。

[监控] 选项卡: [镜像] 选项卡

监控数据库时使用此选项卡。

- **镜像模式** 如果未使用数据库镜像，则显示 [此数据库中没有启用监控]。如果启用镜像，则在使用 -xp 命令行选项所指定的镜像模式为同步时，将会显示 [Synchronous]；否则，会显示 [Asynchronous]。
- **镜像状态** 返回以下各值之一：
 - **正在同步** 未连接镜像服务器或尚未读取主服务器的所有日志页。如果同步模式为异步，也会返回此值。
 - **已同步** 镜像服务器已连接并进行了主服务器上已提交的所有更改。
- **合作伙伴状态** 显示以下各值之一：
 - **已连接** 镜像服务器已连接到主服务器。
 - **断开连接** 镜像服务器未连接到主服务器。
- **仲裁程序状态** 显示以下各值之一：
 - **已连接** 仲裁服务器连接到主服务器。
 - **断开连接** 仲裁服务器未连接到主服务器。

另请参见

- “数据库镜像简介”一节第 880 页

删除旧的监控器度量

您可以自定义监控器保留历史度量的持续时间。可以选择使用部分或全部设置。缺省情况下，监控器会在每天午夜执行一次对其自身的维护操作。维护会影响度量而不影响警告。

◆ **配置对历史度量的删除操作**

1. 单击 [管理]。
2. 单击 [配置] 选项卡。
3. 单击 [编辑]。
4. 单击 [维护]。
5. 指定监控器应执行维护操作的时间。缺省情况下，维护操作在午夜执行。该时间是运行监控器的计算机的本地时间。
6. 自定义 [数据缩减] 设置：
 - **采用超过以下天数的平均值** 选择该选项时，将取超过指定天数的所有数字度量的平均值，然后删除这些数字度量。非数字度量并不删除。
 - **删除超过以下天数的值** 选择该选项时，将删除超过指定时间长度的所有度量。
 - **当 SQL Anywhere 监控器使用的总磁盘空间大于以下值 (MB) 时删除旧值: X** 选择该选项时，需指定可用于存储度量的最大空间量。当使用的磁盘空间数量达到或超过指定数量时，监控器将从最早的度量开始删除，从而防止监控器为其度量使用更多的磁盘空间。删除度量，直到有足够的空间来存储新度量时为止。
7. 单击 [保存]。

管理资源

资源是指 SQL Anywhere 数据库。向监控器添加资源，然后开始监控它们。

缺省资源名为 **SQL Anywhere Monitor**，可报告监控器本身的健康情况。您不能修改该资源，也不能停止对它的监控。

开始监控资源

在您开始监控资源时，监控器便开始收集度量。

对资源的监控将在：

- 添加资源时自动开始。请参见“[添加资源](#)”一节第 935 页。
- 启动监控器时自动开始。缺省情况下，启动监控器时会自动开始对所有现有资源的监控。
- 封锁期结束时自动开始。监控器会自动尝试连接到资源并重新开始监控。
- 当管理员打开 **[管理]** 选项卡时，单击 **[资源]**，从列表中选择一个资源，然后单击 **[启动]**。

添加资源

要监控数据库，您必须首先将资源添加到监控器。

将数据库添加为要监控的资源时，监控器会将对象安装到数据库中来帮助监控器收集数据。添加资源时，必须提供该数据库的 DBA 用户 ID 和口令。这些凭证用于连接到数据库，以及安装监控数据库时所需的数据库对象。然后，DBA 凭证会被放弃。有关已安装的对象列表，请参见“[安装的对象](#)”一节第 950 页。

只有管理员才能添加资源。缺省情况下，添加资源后即会开始对资源进行监控。

◆ 将资源添加到监控器

1. 登录到监控器。
2. 单击 **[管理]** 选项卡。
3. 在 **[资源]** 选项卡上，单击 **[添加]**。
4. 按照 **[添加资源]** 窗口中的说明添加资源来监控数据库。
5. 单击 **[创建]**。

即会添加资源并启动对该资源的监控。

6. 添加 SQL Anywhere 数据库时，必须提供该数据库的 DBA 用户 ID 和口令。这些凭证用于连接到数据库，以及安装监控数据库时所需的数据库对象。然后，DBA 凭证会被放弃。
7. 单击 **[确定]**。

收集间隔

有三种类型的收集间隔：

- **高收集间隔** 该等级用于频繁发生变化的信息，如长时间运行的查询。
- **中收集间隔** 该等级用于变化不太频繁的信息，如可用的磁盘空间量。
- **低收集间隔** 该等级用于很少发生变化的信息，如未提交的错误报告。

管理员可以配置监控器收集资源的度量的频率。为每个资源设置收集间隔。不能配置缺省资源（SQL Anywhere 监控器）。

◆ 编辑收集间隔

1. 单击 **[管理]** 选项卡。
2. 单击 **[资源]** 选项卡，然后从列表中选择一个资源。
3. 单击 **[配置]**。
4. 单击 **[收集间隔]**。
5. 根据需要配置其它设置，然后单击 **[保存]**。
6. 单击 **[确定]**。

另请参见

- [“监控器度量”一节第 928 页](#)
- [“指定要收集的度量”一节第 936 页](#)

指定要收集的度量

管理员可以配置监控器收集哪些度量以及何时应发出警告。不能配置缺省资源（SQL Anywhere 监控器）。

◆ 配置所要收集的度量

1. 单击 **[管理]** 选项卡。
2. 单击 **[资源]** 选项卡，然后从列表中选择一个资源。
3. 单击 **[配置]**。
4. 单击 **[度量]**。选择度量和警告。有关度量和警告的定义，请参见 [“度量和警告的类型。”一节第 937 页](#)。
5. 根据需要配置其它设置。
6. 单击 **[保存]**。

7. 为 SQL Anywhere 数据库编辑资源时，必须提供该数据库的 DBA 用户 ID 和口令。这些凭证用于连接到数据库，以及更改监控数据库所需的已安装数据库对象。然后，DBA 凭证会被放弃。
8. 单击 [确定]。

另请参见

- “[监控器度量](#)”一节第 928 页
- “[收集间隔](#)”一节第 936 页

度量和警告的类型。

下表介绍了 [配置资源] 窗口中可用于资源的度量：[\[度量\]](#) 选项卡。

- **CPU 使用（高收集间隔）** 选择此选项可收集关于数据库 CPU 使用情况的度量。可在 [\[CPU\]](#) 选项卡上查看这些度量。请参见“[\[监控\] 选项卡：\[CPU\] 选项卡](#)”一节第 931 页。
 - **对于一行中的两个收集间隔，当 CPU 使用达到 X% 时发出警告** 当 CPU 使用达到指定的百分比时发出警告。缺省值为 95%。
- **内存使用（高收集间隔）** 选择此选项可收集有关高速缓存的度量。可在 [\[内存\]](#) 选项卡上查看这些度量。请参见“[\[监控\] 选项卡：\[内存\] 选项卡](#)”一节第 931 页。
 - **当内存使用达到 X% 时发出警告** 当资源的内存使用达到指定的百分比时发出警告。缺省值为 85%。
- **磁盘使用（高收集间隔）** 选择此选项可收集关于数据库的 dbspaces 及磁盘读取和写入的度量。可在 [\[磁盘\]](#) 选项卡上查看这些度量。请参见“[\[监控\] 选项卡：\[磁盘\] 选项卡](#)”一节第 932 页。
 - **当每个 dbspace 的可用磁盘空间小于 X 兆字节时发出警告** 当每个 dbspace 的可用磁盘空间小于指定数量时发出警告。缺省值为 100 MB。
- **连接（高收集间隔）** 选择此选项可收集关于连接的度量。可在 [\[连接\]](#) 选项卡上查看这些度量。请参见“[\[监控\] 选项卡：\[连接\] 选项卡](#)”一节第 932 页。
 - **当连接被阻塞的时间大于 X 秒时发出警告** 当连接被阻塞的时间超过指定的时间时发出警告。缺省值是 10 秒。
- **连接计数（中收集间隔）** 选择此选项可收集关于连接数的度量。可在 [\[连接\]](#) 选项卡上查看这些度量。请参见“[\[监控\] 选项卡：\[连接\] 选项卡](#)”一节第 932 页。
 - **当使用中的连接数达到许可限制的 X% 时发出警告** 当使用中的连接数达到许可限制的指定百分比时发出警告。缺省值为 85%。
- **已处理的查询（高收集间隔）** 选择此选项可收集有关查询处理速率的度量。可在 [\[查询\]](#) 选项卡上查看这些度量。请参见“[\[监控\] 选项卡：\[查询\] 选项卡](#)”一节第 933 页。
- **长时间运行查询（中收集间隔）** 选择此选项可收集有关长时间运行的查询的度量。可在 [\[查询\]](#) 选项卡上查看这些度量。请参见“[\[监控\] 选项卡：\[查询\] 选项卡](#)”一节第 933 页。
 - **当一个查询的运行时间超过 X 秒时发出警告** 当一个查询的运行时间超过指定的时间时发出警告。缺省值是 10 秒。

- **失败的连接（中收集间隔）** 选择此选项可收集关于失败连接的度量。可在 [失败的连接] 选项卡上查看这些度量。请参见 “[[监控](#)] 选项卡: [[失败的连接](#)] 选项卡” 一节第 933 页。
- **HTTP 服务器使用（中收集间隔）** 选择此选项可收集关于 HTTP 服务器使用情况的度量。可在 [HTTP] 选项卡上查看这些度量。请参见 “[[监控](#)] 选项卡: [[HTTP](#)] 选项卡” 一节第 932 页。
- **镜像信息（中收集间隔）** 选择此选项可收集关于数据库镜像的度量。可在 [镜像] 选项卡上查看这些度量。请参见 “[[监控](#)] 选项卡: [[镜像](#)] 选项卡” 一节第 933 页。
- **未调度的请求（高收集间隔）** 选择此选项可收集关于未调度的请求的度量。可在 [未调度的请求] 选项卡上查看这些度量。请参见 “[[监控](#)] 选项卡: [[未调度的请求](#)] 选项卡” 一节第 931 页和 “[[监控器疑难解答](#)]” 一节第 952 页。
 - **当未调度的请求数达到以下值时发出警告: X** 当未调度的请求数达到指定的数量时发出警告。缺省值为 5。
- **当在以下环境中发生相同情况时取消发出警告: X 分钟** 选择此选项以阻止在指定时间内接收重复的警告。缺省值为 30 分钟。

停止监控资源

当您不想让监控器从 SQL Anywhere 数据库收集度量时，可停止监控资源。例如，您想要在您知道资源将会不可用时停止监控；否则，您在资源可用前收到警告。除了缺省监控器资源以外，您可在随时停止监控任何资源。

当您停止监控资源时，监控器将：

- 停止为资源收集度量。
- 停止为资源发出警告。

停止监控资源的方法共有两种：

- **调度有规律的、重复的封锁期限** 当符合以下条件时最好选择此方法：
 - 必须反复地停止监控数据库。例如，在每个月的月末执行常规维护。
 - 事先知道数据库不可用的时间。例如，您知道常规维护需要四个小时。
 - 需要监控以自动重启。当封锁结束时，监控器尝试重新连接到资源及继续收集数据。

要使用此方法，可创建封锁来使监控器在指定的时间停止监控。请参见 “[[使用封锁自动停止监控资源](#)]” 一节第 939 页。

- **手工停止监控** 当符合以下条件时最好选择此方法：
 - 需要停止监控很少出现的任务或一次性任务。例如，因为正在运行资源的计算机需要离线才能进行特殊维护，所以您需要停止监控。
 - 可以随后重新启动监控。当资源已手工停止时，监控器等待您重新启动监控。要使用此方法，请参见 “[[手工停止监控资源](#)]” 一节第 939 页。

如果要永久停止监控一个资源，可将它从监控器中删除。请参见 “[[删除资源](#)]” 一节第 940 页。

手工停止监控资源

以下过程介绍如何手工停止资源。有关停止资源时所发生的情况的信息，请参见“[停止监控资源](#)”一节第 938 页。

◆ 手工停止资源

1. 单击 [管理] 选项卡。
2. 选择要停止的资源。
3. 在 [资源] 选项卡上，单击 [停止]。

另请参见

- “[开始监控资源](#)”一节第 935 页
- “[使用封锁自动停止监控资源](#)”一节第 939 页

使用封锁自动停止监控资源

以下过程介绍如使用封锁停止资源。有关停止资源时所发生的情况以及应该使用封锁的时间的信息，请参见“[停止监控资源](#)”一节第 938 页。

封锁是指您不想让监控器收集度量的时间。当封锁结束时，监控器尝试重新连接到资源及继续收集数据。

封锁出现于资源的本地时间。

◆ 配置封锁时间

1. 以管理员身份登录到监控器。
2. 单击 [管理] 选项卡。
3. 在 [资源] 选项卡上，选择要为其指定封锁时间的资源。
4. 单击 [配置]。
5. 单击 [封锁] 选项卡。
6. 单击 [新建]。
7. 在 [新建封锁期] 窗口中，指定封锁的日期和时间。
对于资源数据库所在的计算机，此时间为本地时间。
8. 单击 [保存]。
9. 单击 [保存]。
10. 单击 [确定]。

另请参见

- “开始监控资源”一节第 935 页
- “手工停止监控资源”一节第 939 页

修复数据库资源

如果修复资源，则会重新安装监控资源所需要的数据库对象。监控选项保持不变。

可以只修复 SQL Anywhere 数据库资源。不能修复监控器（名为 SQL Anywhere 监控器）的缺省资源。只有管理员才能修复资源。每次修复资源时，您都必须指定数据库的 DBA 用户 ID 和口令。

◆ 修复 SQL Anywhere 资源

1. 单击 [管理] 选项卡。
2. 单击 [资源] 选项卡。
3. 选择要修复的数据库资源。
4. 如果资源当前正受到监控，请单击 [停止]。
5. 单击 [修复]。
6. 在出现提示时，键入 SQL Anywhere 数据库的 DBA 用户 ID 和口令。这些 DBA 凭证用于连接到数据库，然后系统会放弃这些凭证。
7. 单击 [修复]。
8. 单击 [确定]。
9. 重新启动对资源的监控。请参见“管理资源”一节第 935 页。

删除资源

仅当您确定不需要监控资源时，您才应将其删除；例如，当您不再使用数据库时。

删除资源会使监控器：

- 永久停止监控该资源。
- 丢弃为该资源收集的度量。

当您删除数据库资源时，监控器并不删除那些已安装在数据库中的监控对象。有关删除这些对象的信息，请参见“删除监控对象”一节第 950 页。

只有管理员才能删除资源。不能删除 [SQL Anywhere 监控器] 资源。

◆ 删除资源

1. 单击 [管理] 选项卡。

2. 在 [资源] 选项卡上，选择一个资源，然后单击 [删除]。
3. 单击 [是]。

另请参见

- [“停止监控资源”一节第 938 页](#)

使用监控器用户

监控器支持三类用户：

- **只读用户** 具有监控器资源的只读权限。只读用户可以在 **[监控]** 选项卡上查看度量，但不能访问 **[管理]** 选项卡。需要用户名和口令。
- **运算符** 具有监控器资源的只读权限并可接收警告。这些用户可以在 **[监控]** 选项卡上查看度量，可以接收电子邮件警告，同时可解析和删除警告。但是，操作员不能访问 **[管理]** 选项卡。需要用户名和口令。
- **管理员** 与操作员具有的权限相同，还可以配置资源和添加用户。管理员还可以访问 **[管理]** 选项卡。缺省用户 **[admin]** 是管理员。需要用户名和口令。

用于登录到监控器的用户名和口令区分大小写。

缺省用户

缺省情况下，当您第一次启动监控器时，它有一个管理员用户，用户名为 **admin**，口令为 **admin**。缺省情况下，此用户具有完全权限。建议您更改缺省的管理员口令，以限制对监控器的访问。请参见“[编辑监控器用户](#)”一节第 943 页。

只读访问（无用户名）

缺省情况下，任何用户即使不登录到监控器，也拥有只读权限。但是，出于安全及其它原因，管理员可以要求用户登录。请参见“[要求监控器用户登录](#)”一节第 944 页。

创建监控器用户

您必须是管理员才能添加监控器用户。

◆ 添加新的监控器用户

1. 单击 **[管理]** 选项卡。
2. 单击 **[用户]** 选项卡。
3. 单击 **[新建]**。
4. 填写有关新用户的信息。仅对于应接收来自监控器的电子邮件警告的用户而言，电子邮件地址才为必填项。
单击 **[保存]**。
5. 如果创建了操作员或管理员，可将该用户与资源相关联。请参见“[使监控器用户与资源相关联](#)”一节第 943 页。

另请参见

- “[编辑监控器用户](#)”一节第 943 页

使监控器用户与资源相关联

如果希望用户接收关于关联资源的电子邮件警告，则必须使用户与资源相关联。只能使操作员或管理员与资源相关联。

◆ 使操作员或管理员与资源相关联

1. 单击 **[管理]** 选项卡。
2. 单击 **[资源]** 选项卡。
3. 选择资源，然后单击 **[配置]**。
4. 单击 **[操作员]**。
5. 在 **[可用操作员]** 列表中，选择用户并单击 **[添加]**。
6. 单击 **[保存]**。
7. 单击 **[确定]**。
8. 检查监控器是否已设置为通过电子邮件发送警告通知。请参见“[发送警告电子邮件](#)”一节第 947 页。

另请参见

- “[使用监控器用户](#)”一节第 942 页

编辑监控器用户

作为管理员，您可以编辑监控器用户以更改他们的：

- 口令
- 电子邮件地址
- 语言设置
- 用户类型

◆ 编辑现有监控器用户

1. 单击 **[管理]** 选项卡。
2. 单击 **[用户]** 选项卡。
3. 选择要编辑的用户。
4. 单击 **[编辑]**。
5. 根据需要更改用户的设置。
6. 单击 **[保存]**。

7. 如果正在编辑操作员或管理员，则可将该用户与资源相关联。请参见“[使监控器用户与资源相关联](#)”一节第 943 页。

另请参见

- [“使用监控器用户”一节第 942 页](#)
- [“创建监控器用户”一节第 942 页](#)
- [“删除监控器用户”一节第 944 页](#)

删除监控器用户

删除用户时会将用户从监控器中删除并取消用户与任何资源的关联。

您必须是管理员才能删除监控器用户。

◆ 删除现有监控器用户

1. 单击 **[管理]** 选项卡。
2. 单击 **[用户]** 选项卡。
3. 选择要删除的用户。
4. 单击 **[删除]**。
5. 单击 **[是]** 删除所选用户。单击 **[全部删除]** 删除所有用户。

用户将从监控器中删除。

另请参见

- [“创建监控器用户”一节第 942 页](#)
- [“编辑监控器用户”一节第 943 页](#)
- [“使监控器用户与资源相关联”一节第 943 页](#)

要求监控器用户登录

缺省情况下，任何用户都具有监控器的只读访问权限。您可以更改此行为，以使用户每次在 Web 浏览器中打开监控器时，都必须提供用户名和口令，然后才能看到任何监控数据。

◆ 限制对监控器的访问

1. 单击 **[管理]** 选项卡。
2. 在 **[配置]** 选项卡上，单击 **[编辑]**。
3. 单击 **[验证]**。
4. 清除 **[允许所有人以只读方式访问 SQL Anywhere 监控器]** 选项。
5. 单击 **[保存]**。

另请参见

- [“创建监控器用户”一节第 942 页](#)
- [“编辑监控器用户”一节第 943 页](#)

警告

警告是所关注且应引起管理员或操作员注意的某一条件或状态。警告包括有关问题起因的信息，并提供有关如何解决问题的建议。

对于诸如磁盘空间少、重要软件更新、登录尝试失败以及内存利用率高之类的条件，有多种预定义的警告。当满足某个警告条件时，**[监控]** 选项卡上的底部窗格会列出该警告。在顶部窗格中，数据库的**[状态]** 将发生变化来指示存在警告。您可以对监控器进行配置，使其在出现警告时向操作员和管理员发送电子邮件。请参见“[发送警告电子邮件](#)”一节第 947 页。

监控器基于所收集的度量对警告进行检测。而非在受监控的数据库上检测警告。您可更改缺省阈值，并通过编辑资源来选择启用哪些警告。请参见“[监控器度量](#)”一节第 928 页。

查看警告

任何用户都可查看警告；但是，只有操作员和管理员才能解析和删除警告。

◆ 查看警告

1. 单击 **[监控]** 选项卡。
2. 从列表中选择一個资源。
3. 在底部窗格中，单击 **[警告]** 选项卡。
4. 在警告列表中选择一行。
5. 单击 **[详细信息]**。
6. 单击 **[确定]**。

另请参见

- “[解析警告](#)”一节第 946 页
- “[删除警告](#)”一节第 947 页
- “[发送警告电子邮件](#)”一节第 947 页

解析警告

一旦触发警告的问题得以解决，即可将警告标记为已解析。解析警告会致使监控器更改警告的状态列，但警告仍在警告列表中。如果要移除该警告，必须将它删除。请参见“[删除警告](#)”一节第 947 页。

只有操作员和管理员才能解析警告。

◆ 解析警告

1. 单击 **[监控]** 选项卡。
2. 从列表中选择一個资源。

3. 在底部窗格中，单击 **[警告]** 选项卡。
4. 在警告列表中选择该行。
5. 单击 **[标记已解决]** 解析所选的警告。单击 **[标记已全部解决]** 解析列表中的所有警告。
在 **[警告]** 选项卡上 **[状态]** 列中的值更改为 **[已解决]**。
如果此警告是这个资源唯一的未解析警告，则资源的状态会更改为 **[良好]**。

另请参见

- “删除警告”一节第 947 页
- “解析警告”一节第 946 页
- “发送警告电子邮件”一节第 947 页
- “查看警告”一节第 946 页
- “警告”一节第 946 页

删除警告

在警告列表中监控器只保留最新的 50 个警告。如果不希望某个警告再出现在警告列表中，可将其删除。您可以删除警告，而无论其状态如何。

只有操作员和管理员才可以删除警告。

◆ 删除警告

1. 单击 **[监控]** 选项卡。
2. 从列表选择一个资源。
3. 在底部窗格中，单击 **[警告]** 选项卡。
4. 在警告列表选择一个行。
5. 单击 **[删除]**。

警告即会从警告列表中删除。

另请参见

- “解析警告”一节第 946 页
- “发送警告电子邮件”一节第 947 页
- “查看警告”一节第 946 页
- “警告”一节第 946 页

发送警告电子邮件

您可以对监控器进行配置，使其在出现警告时向操作员和管理员发送电子邮件。

要使监控器通过电子邮件发送警告通知，您必须：

1. 创建具有电子邮件地址的管理员或操作员。请参见“[创建监控器用户](#)”一节第 942 页。
2. 使管理员或操作员与某个资源相关联。请参见“[使监控器用户与资源相关联](#)”一节第 943 页。
3. 使监控器能够发送电子邮件。请参见“[使监控器能够发送警告电子邮件](#)”一节第 948 页。

使监控器能够发送警告电子邮件

作为管理员，您可以配置监控器，使其在出现警告时发送电子邮件。监控器支持用于发送电子邮件的 SMTP 和 MAPI 协议。

◆ 使监控器能够通过电子邮件发送警告通知

1. 单击 [管理] 选项卡。
2. 单击 [配置] 选项卡。
3. 单击 [编辑]。
4. 单击 [警告通知]。
5. 选择 [通过电子邮件发送警告通知]。
6. 为 [您要使用哪个协议来通过电子邮件发送警告?] 字段选择 SMTP 或 MAPI。
7. 根据需要配置其它设置。

● MAPI

- 用户名** 键入 MAPI 服务器的用户名。
- 口令** 键入 MAPI 服务器的口令。

● SMTP

- 服务器** 指定要使用哪个 SMTP 服务器。键入 SMTP 服务器的服务器名或 IP 地址。例如 *SMTP.yourcompany.com*。
- 端口** 指定连接至 SMTP 服务器的端口编号。缺省值为 25。
- 发送者姓名** 为发送者的电子邮件地址指定别名。例如 *JoeSmith*。
- 发送者地址** 指定发送者的电子邮件地址。例如 *jsmith@emailaddress.com*。
- 此 SMTP 服务器要求验证** 如果 SMTP 服务器需要验证，则选择此选项。
 - **用户名** 指定要提供给需要验证的 SMTP 服务器的用户名。
 - **口令** 指定要提供给需要验证的 SMTP 服务器的口令。

8. 测试您是否已正确配置了电子邮件通知。
单击 [发送测试电子邮件]。
9. 在系统提示时，输入要将测试电子邮件发送到的电子邮件地址，然后单击 [确定]。
测试电子邮件将会被发送到指定的电子邮件地址。
10. 单击 [保存]。

另请参见

- “解析警告”一节第 946 页
- “删除警告”一节第 947 页
- “查看警告”一节第 946 页

取消对资源的未提交错误报告的警告

作为管理员，您可以配置在资源有未提交的错误报告时监控器是否发出警告。缺省情况下，监控器并不发送这些警告。有关错误报告及其提交方式的信息，请参见“[SQL Anywhere 中的错误报告](#)”一节第 76 页。

◆ 取消对未提交的错误报告的警告

1. 单击 [管理] 选项卡
2. 单击 [配置] 选项卡。
3. 单击 [编辑]。
4. 单击 [选项]。
5. 单击 [保存]。

安装的对象

下表列出了监控 SQL Anywhere 数据库时安装的对象。

对象名称	对象类型	说明
sa_monitor_user	数据库用户	此只读用户被添加到数据库中以收集度量。由于此用户被添加到正在监控的数据库中，因此不必将 DBA 凭证存储在所监控数据库以外的任何位置。可能有必要允许 sa_monitor_user 跳过口令验证。sa_monitor_user 具有只有监控器知道的随机口令，且其不具有管理员特权。
sa_monitor_connection_failure	表	该表包含有关失败链接尝试的度量，它与 sa_monitor_connection_failed_event 一起使用。当从监控器检索度量时，此表中的度量会被删除。
sa_monitor_connection_failed_event	事件	该事件在出现 ConnectFailed 系统事件时（每次连接尝试失败时）触发，并将一条记录插入到 sa_monitor_connection_failure 表中。
sa_monitor_count_unsubmitted_crash_reports	功能	该函数调用 xp_srvmon_count_unsubmitted_crash_reports 过程，以求得未提交的崩溃报告的总数。

删除监控对象

因为数据库对象是由单个所有者所拥有，所以您可以通过执行以下语句将其全部删除。

```
DROP USER sa_monitor_user;
```

重新安装监控对象

要重新安装数据库对象，请参见“[修复数据库资源](#)”一节第 940 页。

在一台单独的计算机上安装 SQL Anywhere 监控器

以下说明将介绍如何在未运行 SQL Anywhere 的另外一台单独计算机上安装 SQL Anywhere 监控器。

在单独的计算机上运行 SQL Anywhere 监控器具有以下优势：

- 监控器可以在后台作为服务来运行。
- 计算机启动时，监控器将自动启动。
- 如果监控器安装在单独的计算机上，SQL Anywhere 的升级和更新不会覆盖监控器。当该单独的计算机处于生产环境中时，这一点很重要。

◆ 在单独的计算机上安装监控器

- 运行安装介质的 *Monitor* 目录下的 *setup.exe* 文件，并遵照提供的说明进行安装。

监控器疑难解答

问题	建议
在您按 F5 键刷新 Web 浏览器窗口时，您需要登录到监控器。	在 Web 浏览器中启用 JavaScript。
在尝试连接到监控器时收到网络通信错误。	启动监控器。请参见“启动监控器”一节第 923 页。
在升级到 Adobe Flash Player 的最新版本后，继续接收说明以便可以持续升级 Adobe Flash Player。	验证您的操作系统是否支持已安装的 Adobe Flash Player 版本。监控器向后兼容 Adobe Flash Player 版本 9。要确定正确的版本，请访问： http://www.adobe.com/products/flashplayer/systemreqs/ 。
监控器无法开始监控 SQL Anywhere 数据库资源。	检查该资源的口令验证功能和登录过程是否允许用户 sa_monitor_user 连接到该资源。
未收到任何警告电子邮件。	<p>检查监控器是否已被正确配置为发送电子邮件及发送测试电子邮件。请参见“使监控器能够发送警告电子邮件”一节第 948 页。</p> <p>检查来自监控器的警告电子邮件是否未被病毒扫描程序拦截。请参见“xp_startsmtp 系统过程”一节《SQL Anywhere 服务器 - SQL 参考》。</p>
监控器所报告的未调度的请求数似乎少于实际的未调度请求数。	<p>当收集有关未调度请求数的度量时，监控器执行了关于资源的查询。此查询可能是未调度的请求。</p> <p>在未调度的查询出现时依序对其进行处理。因此，当监控器尝试执行其查询时，如果存在未调度的请求，则此查询必须等待现有未调度的请求完成后，才能够加以执行。</p> <p>因此，当监控器收集未调度的请求数时，此数量并不包括监控器发出其查询的时间与请求被执行的时间之间所存在的未调度的请求。</p>
当数据库磁盘空间超过指定阈值时，您没有收到警告。	<p>在监控器的收集间隔之间，数据库可以超过指定的磁盘空间警告阈值和可用的空间大小。在这种情况下，数据库将会在监控器能够收集磁盘使用情况度量并发出警告之前停止响应。</p> <p>如果数据库快速增大，请将磁盘空间警告阈值设置为较大的数值，以便您能够在数据库运行空间不足之前收到警告。请参见“度量和警告的类型。”一节第 937 页。</p>

问题	建议
<p>在“非英文”计算机的 Firefox Web 浏览器中打开监控器时，监控器将以英文显示。</p>	<p>Firefox 无法正确使用您的计算机上的首选区域设置。您可以使用 Internet Explorer，或在 Firefox 中尝试以下的变通解决方法：</p> <ol style="list-style-type: none"> 1. 在 Firefox 中打开一个新选项卡。 2. 在地址栏中，键入以下内容： <pre style="margin-left: 40px;">about:config</pre> <p>按 Enter 键。</p> <p>出现提示时，单击 [I'll Be careful, I Promise!]</p> 3. 在 [Filter] 字段中，键入以下内容： <pre style="margin-left: 40px;">general.useragent.locale</pre> 4. 在首选项列表中，双击 [general.useragent.locale]。 5. 在 [Enter String Value] 窗口中，输入您的区域设置。例如，键入 fr-FR 表示法语；键入 de-DE 表示德语；键入 zh-CN 表示简体中文；键入 ja-JP 表示日语。 6. 单击 [OK]。

SQL Anywhere SNMP Extension Agent

目录

SQL Anywhere SNMP Extension Agent 简介	956
了解 SNMP	957
使用 SQL Anywhere SNMP Extension Agent	961
SQL Anywhere MIB 参考	969
RDBMS MIB 参考	995

SQL Anywhere SNMP Extension Agent 简介

如果您正在 Windows（32 位版本）上运行 SQL Anywhere，可以将 SQL Anywhere SNMP Extension Agent 与 SNMP 管理应用程序结合使用，以管理 SQL Anywhere 数据库。一个代理可以用来监控在不同数据库服务器（在不同计算机上运行）上运行的多个不同的数据库。

使用 SQL Anywhere SNMP Extension Agent，您能够：

- 检索所有服务器统计信息和数据库统计信息的值。
- 检索所有服务器属性和数据库属性的值。
- 检索所有 PUBLIC 数据库选项的值。
- 设置任何 PUBLIC 数据库选项的值。
- 执行存储过程。
- 生成基于属性值或统计信息值的陷阱。

支持的文件

SQL Anywhere SNMP Extension Agent 的以下文件包含在 SQL Anywhere 安装中：

- **dbsnmp11.dll** SQL Anywhere SNMP Extension Agent。此文件位于 *install-dir\bin32* 中。
- **iAnywhere.mib** SQL Anywhere MIB 包含可以使用 SQL Anywhere SNMP Extension Agent 访问的数据库服务器和数据库属性、统计信息和选项的所有 OID。
- **RDBMS-MIB.mib** 它是关系数据库管理系统的通用 MIB，其中包含可以使用 SQL Anywhere SNMP Extension Agent 访问的 OID。
- **SNMPv2-SMI.mib** 此 MIB 由 SQL Anywhere 和 RDBMS MIB 引用。
- **SNMPv2-TC.mib** 此 MIB 由 SQL Anywhere 和 RDBMS MIB 引用。
- **SYBASE-MIB.mib** Sybase MIB。此 MIB 由 SQL Anywhere MIB 引用。
- **sasnm.ini** 此文件列出了 SQL Anywhere SNMP Extension Agent 监控的数据库。缺省情况下，此文件位于 *install-dir\bin32* 中。

了解 SNMP

简单网络管理协议（Simple Network Management Protocol，简称 **SNMP**）是用于网络管理的标准协议。SNMP 允许**管理器**和**代理**进行通信：管理器向代理发送请求，代理响应来自管理器的查询。此外，在发生特定事件时，代理还可以使用通知（称为**陷阱**）来通知管理器。

SNMP 代理处理用以获取和设置托管对象变量的值的请求。每个变量都有一个值，这些值通常为字符串或整数，当然也可以是其它类型。

变量保存在全局层次中，每个变量在其父级下都有一个唯一编号。变量的完整名称（包括它的所有父级）称为**对象标识符**（Object Identifier，简称 **OID**）。Sybase 拥有的所有 OID 都以

1.3.6.1.4.1.897 开头。

代理支持的 OID（包括它们的名称、类型和其它信息）的列表存储在一个名为**管理信息库**（Management Information Base，简称 **MIB**）的文件中。

MIB 是一个数据库，其中存储关于托管对象的网络管理信息。MIB 不同于使用 SQL Anywhere SNMP Extension Agent 监控的 SQL Anywhere 数据库。MIB 对象的值可以使用 SNMP 来更改或检索。MIB 对象以层次组织，网络的最一般信息位于层次顶层。SQL Anywhere SNMP Extension Agent 支持以下 MIB：

- **SQL Anywhere MIB** 专门为 SQL Anywhere SNMP Extension Agent 创建的 MIB。SQL Anywhere MIB 中的所有 OID 都以 **1.3.6.1.4.1.897.2** 开头。SQL Anywhere MIB 列出了可以使用 SQL Anywhere SNMP Extension Agent 检索和设置（在某些情况下）的统计信息值、属性值和选项值的 OID。请参见“[SQL Anywhere MIB](#)”一节第 957 页。
- **RDBMS MIB** 关系数据库的通用 MIB，它与供应商无关。此 MIB 包含关于系统中的数据库服务器和数据库的信息。请参见“[RDBMS MIB](#)”一节第 959 页。

SQL Anywhere MIB

SQL Anywhere MIB 是为 SQL Anywhere SNMP Extension Agent 创建的。它包含所有数据库服务器统计信息和属性，以及所有数据库统计信息、属性和选项。统计信息和属性都是只读的（除了少数几个例外），而数据库选项都是可读写的。

缺省情况下，SQL Anywhere MIB 位于 `install-dir\snmp\iAnywhere.mib` 中。

有关 SQL Anywhere MIB 中表的详细信息，请参见“[SQL Anywhere MIB 参考](#)”一节第 969 页。

有关在 SQL Anywhere MIB 中设置值的详细信息，请参见“[使用 SQL Anywhere SNMP Extension Agent 设置值](#)”一节第 965 页。

以下层次介绍了 SQL Anywhere MIB：

OID	名称	说明
1.3.6.1.4.897.2.1.1.n.db	saServer.saSrvStat	返回数据库 <i>db</i> 上的服务器统计信息 <i>n</i> 的值。

OID	名称	说明
1.3.6.1.4.897.2.1.2. <i>n.db</i>	saServer.saSrvProp	返回数据库 <i>db</i> 上的服务器属性 <i>n</i> 的值。
1.3.6.1.4.897.2.2.1. <i>n.db</i>	saDb.saDbStat	返回数据库 <i>db</i> 上的数据库统计信息 <i>n</i> 的值。
1.3.6.1.4.897.2.2.2. <i>n.db</i>	saDb.saDbProp	返回数据库 <i>db</i> 上的数据库属性 <i>n</i> 的值。
1.3.6.1.4.897.2.2.3. <i>n.db</i>	saDb.saDbOpt	返回数据库 <i>db</i> 上的数据库选项 <i>n</i> 的值。
1.3.6.1.4.897.2.3.1	saAgent.saVersion	返回 SQL Anywhere Extension Agent 的版本。
1.3.6.1.4.897.2.3.2. <i>db</i>	saAgent.saDbConnStr	返回数据库 <i>db</i> 的连接字符串。
1.3.6.1.4.897.2.3.3. <i>db</i>	saAgent.saConnected	返回 SQL Anywhere Extension Agent 是否连接到数据库 <i>db</i> 。将此值设置为 0，将导致 SQL Anywhere Extension Agent 断开与数据库的连接；而将此值设置为 1，则导致 SQL Anywhere Extension Agent 尝试连接到数据库。
1.3.6.1.4.897.2.3.4. <i>db</i>	saAgent.saStarted	返回数据库 <i>db</i> 是否正在运行。将此值设置为 0，将导致 SQL Anywhere Extension Agent 关闭数据库 ¹ ；而将此值设置为 1，则尝试启动数据库 ² 。
1.3.6.1.4.897.2.3.5. <i>db</i>	saAgent.saProc	将此值设置为字符串 <i>proc_name</i> 将导致 SQL Anywhere Extension Agent 执行数据库中的过程 <i>proc_name</i> 。 可以提供参数（例如， <i>proc_name</i> ('string',4)）；如果未提供参数，则在名称后附加括号 0。获取该值将返回 ""。
1.3.6.1.4.897.2.3.6	saAgent.saRestart	将此变量的值设置为 1 导致代理重新启动自己（它与所有数据库断开连接，然后重新装载 <i>.ini</i> 文件）。获取该值将返回 0。

OID	名称	说明
1.3.6.1.4.897.2.3.7	saAgent.saInifile	返回 SQL Anywhere Extension Agent 所使用的 <i>sasnmplib.ini</i> 文件的完整路径。
1.3.6.1.4.897.2.4	saMetaData	几个虚拟表；每一行表示一个由 SQL Anywhere MIB 支持的变量。

¹ 通过设置此变量来停止数据库时，该停止是无条件的，这意味着即使有活动连接，数据库也会停止。

² 为了能够通过设置此变量来启动数据库，必须在连接字符串中指定 DBF 参数（如果需要，则需要包括 DBN 和 DBKEY），并且还必须在 *sasnmplib.ini* 文件中设置 UtilDbPwd 字段或将服务器的启动数据库权限（用 -gd 服务器选项指定）设置为 all。

saMetaData 表

SQL Anywhere MIB 包含元数据表，它们提供了对 SQL Anywhere Extension Agent 进行查询从而找出所支持变量的方式。

- **saSrvMetaData.saSrvStatMetaDataTable** 列出数据库服务器统计信息（sa.saServer.saSrvStat 下的变量）。
- **saSrvMetaData.saSrvpropMetaDataTable** 列出数据库服务器属性（sa.saServer.saSrv.Pro 下的变量）。
- **saDbMetaData.saDbStatMetaDataTable** 列出数据库统计信息（sa.saDb.saDbStat 下的变量）。
- **saDbMetaData.saDbpropMetaDataTable** 列出数据库属性（sa.saDb.saDbProp 下的变量）。
- **saDbMetaData.saDbOptMetaDataTable** 列出数据库选项（sa.saDb.saDbOpt 下的变量）。

有关存储在 SQL Anywhere MIB 元数据表中信息的详细信息，请参见“[saMetaData 表](#)”一节第 969 页。

RDBMS MIB

RDBMS MIB 是用于关系数据库管理系统产品的、独立于供应商的通用 MIB (RFC 1697)。RDBMS MIB 使用**虚拟表**返回服务器和数据库上的信息。基 OID 为 **1.3.6.1.2.1.39**，在此 MIB 中有 9 个虚拟表。SQL Anywhere SNMP Extension Agent 支持其中 8 个虚拟表。

有关 RDBMS MIB 中包含的表的详细信息，请参见“[RDBMS MIB 参考](#)”一节第 995 页。

SQL Anywhere Extension Agent 提供对 RDBMS MIB 中所有支持的变量的只读访问权限。RDBMS MIB 中的变量都不能通过 SQL Anywhere Extension Agent 写入。

虚拟表包含固定数目的属性和任意数目的行。通过将列号和行号附加到表的 OID，使用 GET 请求可检索表中的元素。必须将 1 附加到表 OID，因此 OID 具有如下形式：

`table.1.column.rownum`

缺省情况下，RDBMS MIB 位于 *install-dir\snmp\RDBMS-MIB.mib* 中。

使用 SQL Anywhere SNMP Extension Agent

要使用 SQL Anywhere SNMP Extension Agent，必须在计算机上安装 SNMP 且必须创建一个 *sasnmplib.ini* 文件，其中包含由 SQL Anywhere SNMP Extension Agent 监控的数据库的相关信息。

安装 SNMP

必须在计算机上安装了 SNMP，才能使用 SQL Anywhere Extension Agent。缺省情况下，在 Windows 上不安装 SNMP。

有关安装 SNMP 的信息，请参见相应的操作系统文档。

一旦在计算机上安装了 SNMP，计算机上应运行以下服务：SNMP Service 和 SNMP Trap Service。

如果 SNMP 是在 SQL Anywhere 之前安装的，则需要停止并重新启动 SNMP 服务，以使改服务能够检测到 SQL Anywhere SNMP Extension Agent。如果是先安装 SQL Anywhere 再安装 SNMP，则 SNMP 服务将自动检测 SQL Anywhere SNMP Extension Agent。

◆ 重新启动 SNMP 服务（命令行）

1. 运行以下命令：

```
net stop snmp
```

这将停止 SNMP 服务。

2. 运行以下命令：

```
net start snmp
```

这将启动 SNMP 服务。

配置 SQL Anywhere SNMP Extension Agent

SQL Anywhere Extension Agent 可以监控一个或多个数据库。要被监控的数据库按照以下格式存储在 *sasnmplib.ini* 文件中：

```
[SAAgent]
TrapPollTime=time-in-seconds

[DBn]
ConnStr=connection-string
UtilDbPwd=utility-database-password
CacheTime=time-in-seconds
DBSpaceCacheTime=time-in-seconds
Trap=trap-information
Disabled=1 or 0
```

缺省情况下，SQL Anywhere 安装将 *sasnmplib.ini* 文件放在 *install-dir\bin32* 目录中。

SAAgent 部分

文件 *sasnmplib.ini* 的 SAAgent 部分包含有关 SQL Anywhere Extension Agent 的信息。如果不需要 TrapPollTime 字段，则可以忽略该部分。

TrapPollTime 此值指定对动态陷阱（如果指定）的轮询频率。缺省情况下，SQL Anywhere SNMP Extension Agent 每 5 秒钟对这些值轮询一次。将此值设置为 0 则禁用动态陷阱。此字段为可选字段。

DBn 部分

sasnmplib.ini 文件的每个 **DBn** 部分都描述一个数据库，如何连接到该数据库以及该数据库中存在的的所有动态陷阱。此部分中的字段区分大小写。

n 的值是用于标识数据库的编号。编号必须从 1 开始，并且不能跳过任何编号。例如，如果 *sasnmplib.ini* 文件包含 [DB1]、[DB2] 和 [DB4] 条目，则会忽略 [DB4] 条目，原因是文件中缺少 [DB3] 条目。

ConnStr 用于连接到数据库的连接字符串。您必须提供足够的信息才能连接到数据库。此字段为必填字段。

- 如果要使用 ODBC 数据源连接到数据库，则 ODBC 数据源必须是系统数据源，而不能是用户数据源。
- 如果要使用集成登录，则必须映射到 SYSTEM 帐户，因为 SNMP Agent 是作为服务运行的。但是，这意味着任何作为服务运行的程序无需口令都可以连接到数据库。或者，也可以更改服务运行时使用的帐户，然后为该帐户创建集成登录。
- ASTART=NO; IDLE=0; CON=SNMP; ASTOP=NO 字符串被放到连接字符串的前面。此字符串会起到以下作用：
 - 防止 SQL Anywhere SNMP Extension Agent 试图自动启动数据库服务器
 - 禁用空闲超时，因为 SQL Anywhere SNMP Extension Agent 在一段时间内处于空闲状态的情况很可能会发生
 - 命名连接以便能够标识
 - 防止数据库在 SQL Anywhere SNMP Extension Agent 断开连接时被关闭

如果在 *sasnmplib.ini* 文件中指定了连接字符串中的任意值，则 *sasnmplib.ini* 文件中的值会替换缺省设置。

UtilDbPwd 设置 sa.agent.saStarted 来启动数据库时，SQL Anywhere SNMP Extension Agent 会试图使用 DBF 参数（该参数会将在何处查找数据库文件告诉给数据库服务器）连接到数据库。但是，如果启动数据库需要的权限是 DBA（网络服务器的缺省值，个人服务器和网络服务器都可以使用 -gd dba 选项来进行设置），则服务器将不允许该连接。

要启动此类服务器上的数据库，SQL Anywhere SNMP Extension Agent 必须作为具有 DBA 权限的用户连接到已在同一服务器上运行的数据库。这可以通过连接到实用程序数据库来实现。如果在 *sasnmplib.ini* 文件中指定实用程序数据库口令（由 -su 服务器选项指定），然后启动数据库，则 SQL Anywhere Extension Agent 会连接到同一服务器上的实用程序数据库，执行 START DATABASE 语句，然后断开连接。此字段为可选字段。

CacheTime 从数据库中检索到的数据可以高速缓存到 SQL Anywhere SNMP Extension Agent 中，以便在对同一类型的数据（如服务器属性或数据库统计信息）进行后续检索时无需与数据库进行通信。虽然高速缓存数据意味着可以在后续检索时更快地获取数据，但数据可能会过时。**CacheTime** 字段可用于更改高速缓存时间或通过将该值设置为 0 来禁用高速缓存。缺省情况下，高速缓存时间为 0 秒。如果 **CacheTime** 参数设置为 0，则检索到的数据总是最新的，因为每次请求的数据都是从数据库中检索的。此字段为可选字段。

DBSpaceCacheTime RDBMS MIB 中的 `rdbmsDbLimitedResourceTable` 表包含有关 `dbspace` 的信息。从数据库中检索到的此信息也可以高速缓存在 SQL Anywhere Extension Agent 中。`dbspace` 信息的缺省高速缓存时间为 600 秒（10 分钟）。此字段可用于更改高速缓存时间（或通过将该值设置为 0 来禁用高速缓存）。此字段为可选。请参见“[rdbmsDbLimitedResourceTable](#)”一节第 997 页。

Trap 创建动态陷阱。值 *t* 必须是从 1 开始的正整数。不允许跳过编号。此字段为可选字段。请参见“[创建动态陷阱](#)”一节第 966 页。

Disabled 如果设置为 1，则 SQL Anywhere SNMP Extension Agent 将跳过此数据库条目。这对于从 SQL Anywhere SNMP Extension Agent 管理的数据库列表中临时删除一个数据库而言非常有用，这时无需对其余条目重新编号。此字段为可选字段。

对此文件进行编辑后，必须重新启动 SNMP 服务或重置 SQL Anywhere SNMP Extension Agent，以便 Agent 使用新设置。

◆ 重新启动 SNMP 服务（命令行）

1. 运行以下命令：

```
net stop snmp
```

这将停止 SNMP 服务。

2. 执行以下命令：

```
net start snmp
```

这将启动 SNMP 服务。

◆ 重新启动 SQL Anywhere SNMP Extension Agent

- 使用 SNMP 管理工具，将 `saAgent.saRestart` 属性的值 `1.3.6.1.4.1.897.2.3.6` 更改为 `1`。

可以使用文件隐藏实用程序 (`dbfhide`) 通过简单加密的方式对 `sasnmplib.ini` 文件的内容进行模糊处理。请参见“[隐藏 .ini 文件的内容](#)”一节第 724 页。

sasnmplib.ini 文件示例

以下是 SQL Anywhere SNMP Extension Agent 的 `sasnmplib.ini` 文件的一个示例。

```
[SAAgent]
[DB1]
ConnStr=UID=DBA;PWD=sql;ENG=server1;DBN=sales;DBF=sales.db
Trap1=1.1.5 > 50000
UtilDbPwd=test
[DB2]
```

```

ConnStr=UID=DBA;PWD=sql;ENG=server1;DBN=field;DBF=field.db
UtilDbPwd=test
Disabled=1
[DB3]
ConnStr=UID=DBA;PWD=sql;LINKS=tcPIP;ENG=server2;DBN=hq;DBF=hq.db
UtilDbPwd=test

```

由于在 SAAgent 部分没有指定任何参数，SQL Anywhere SNMP Extension Agent 将每 5 秒钟轮询一次值。

SQL Anywhere SNMP Extension Agent 正在监控在两个不同服务器上运行的 3 个不同的数据库。数据库 3 在不同的计算机上运行，因此需要 LINKS 连接参数来指定协议。为 DB1 指定了一个陷阱，它将在数据库服务器发送的字节数大于 50000 时触发。

使用 SQL Anywhere SNMP Extension Agent 获取值

使用 SQL Anywhere SNMP Extension Agent 可以检索以下各项的值：

- 数据库服务器属性。请参见“SQL Anywhere MIB 服务器属性”一节第 975 页。
- 数据库服务器统计信息。请参见“SQL Anywhere MIB 服务器统计信息”一节第 972 页。
- 数据库选项。请参见“SQL Anywhere MIB 数据库选项”一节第 987 页。
- 数据库属性。请参见“SQL Anywhere MIB 数据库属性”一节第 983 页。
- 数据库统计信息。请参见“SQL Anywhere MIB 数据库统计信息”一节第 980 页。

检索这些值的方式要视 SNMP 管理软件而定。

示例

下表提供了下列 OID 的说明及可能返回的示例值。

OID	解释	示例值
1.3.6.1.4.1.897.2.1.1.1.1	数据库 1 上的服务器统计信息 ActiveReq	1
1.3.6.1.4.1.897.2.2.1.4.1	数据库 1 上的数据库统计信息 CacheRead	11397
1.3.6.1.4.1.897.2.3.1	Agent 版本	11.0.1(2459)
1.3.6.1.4.1.897.2.3.2.1	数据库 1 的连接字符串	UID=DBA;PWD=sql; ENG=server1;DBN=sales; DBF=sales.db

使用 SQL Anywhere SNMP Extension Agent 设置值

SQL Anywhere SNMP Extension Agent 响应 SNMP get、get-next 和 set 查询。

可以使用 SQL Anywhere SNMP Agent 设置任一数据库选项、部分服务器属性以及一个数据库属性。

设置数据库选项时，SQL Anywhere SNMP Agent 执行以下语句：

```
SET OPTION PUBLIC.option-name = 'value'
```

设置数据库和服务器属性时，会使用 sa_server_option 系统过程。

设置这些值的方式要视 SNMP 管理软件而定。

有关可以使用 SQL Anywhere SNMP Extension Agent 设置的选项和属性的详细信息，请参见“SQL Anywhere MIB 参考”一节第 969 页。

另请参见

- “SET OPTION 语句”一节《SQL Anywhere 服务器 - SQL 参考》
- “数据库选项简介”一节第 462 页
- “sa_server_option 系统过程”一节《SQL Anywhere 服务器 - SQL 参考》

使用 SQL Anywhere SNMP Extension Agent 执行存储过程

SQL Anywhere MIB 包含一个 OID，它允许您使用 SQL Anywhere SNMP Extension Agent 来执行存储过程。要执行存储过程，SQL Anywhere SNMP Extension Agent 用来连接的用户必须满足以下条件之一：

- 具有该过程的执行权限
- 是该过程的所有者
- 具有 DBA 权限

将忽略过程生成的所有结果集或返回值。

要使用 SQL Anywhere SNMP Extension Agent 执行存储过程，请将 saAgent.saProc 的值（OID 1.3.6.1.4.1.897.2.3.5.db，其中 db 为 sasnmplib.ini 文件中的数据库编号）设置为代表存储过程名称的字符串。或者，可以向过程提供参数；如果未提供任何参数，则将括号附加到过程名称上。

例如，将 saAgent.saProc 的值设置为字符串 pchin.updatesales('param1', 2) 将调用用户 pchin 拥有的 updatesales 存储过程。

将此 OID 的值设置为过程名的方式要视 SNMP 管理软件而定。请参见“SQL Anywhere MIB”一节第 957 页。

使用陷阱

陷阱是 SNMP 代理在发生特定事件时所发送的 OID。陷阱由 SNMP 代理启动，SNMP 管理软件在检测到该陷阱后，会直接处理该事件或查询 SNMP 代理以获取详细信息。

要接收陷阱，必须配置 SNMP 服务。SNMP 服务将接收陷阱信息再将其转发到某处；但缺省情况下并不转发到任何地方，因此运行中的所有陷阱监听器均检测不到任何陷阱。以下步骤介绍如何配置 SNMP Service，使其向您的计算机发送陷阱。

◆ 配置 SNMP Service

1. 右击 [我的电脑]，然后选择 [管理]。
2. 在左窗格中，双击 [服务和应用程序]。
3. 在左窗格中，双击 [服务]。
4. 在右窗格中的服务列表中找到 SNMP Service，右击并选择 [属性]。
5. 单击 [陷阱] 选项卡。
6. 单击 [添加]。
7. 在 [SNMP Service 配置] 窗口中，在文本框中键入 localhost，然后单击 [添加]。
8. 单击 [确定]。

SQL Anywhere SNMP Extension Agent 陷阱

每当数据库服务器删除连接时，SQL Anywhere SNMP Extension Agent 都会发送一个陷阱。此陷阱的 OID 为 1.3.6.1.2.1.39.2.1。

如果您正在使用数据库镜像，而 SQL Anywhere SNMP Extension Agent 与数据库服务器之间的连接被删除，则 SQL Anywhere SNMP Extension Agent 会每隔 30 秒钟尝试一次与数据库服务器之间重新建立连接。代理重新建立连接后，如果发现连接的是另一个数据库服务器（由 ServerName 属性来确定），则它会从 *sasnmplib.ini* 文件发送一个 OID 为 1.3.6.1.4.1.897.2.6.3 的陷阱和此数据库的 ID。这时，SQL Anywhere SNMP Extension Agent 之前连接到的主数据库服务器已出现故障，现在由镜像服务器担当主服务器。请参见“数据库镜像简介”一节第 880 页。

由 SQL Anywhere SNMP Extension Agent 发送的唯一其它陷阱是动态陷阱。请参见“创建动态陷阱”一节第 966 页。

创建动态陷阱

动态陷阱 是当涉及特定属性值、统计信息值或选项值的简单表达式为 true 时由 SQL Anywhere Extension Agent 发送的陷阱。动态陷阱在 *sasnmplib.ini* 文件中创建。*sasnmplib.ini* 文件条目中陷阱信息的格式如下所示：

```
Traptrapnum=[1.3.6.1.4.1.897.2.]oid[.dbnum] op value
```

trapnum 是动态陷阱编号。它必须从 1 开始并且是连续的。

oid 是属性、统计信息或选项的 OID。SQL Anywhere MIB 或 RDBMS MIB 中的 OID 会得到支持。如果给定的 OID 是无效的 SQL Anywhere OID 或 RDBMS OID，则会将 SQL Anywhere MIB 前缀 (1.3.6.1.4.1.897.2.) 放在前面。

有关 SQL Anywhere MIB 中 OID 的信息，请参见“SQL Anywhere MIB 参考”一节第 969 页。

有关 RDBMS MIB 中 OID 的信息，请参见“[RDBMS MIB 参考](#)”一节第 995 页。

注意

您只能使用与数据库服务器或动态陷阱中的数据库属性、统计信息或选项相对应的 OID。

dbnum 是数据库编号。此字段为可选字段，但如果指定，则必须与 *sasnmplib.ini* 文件的 [DBn] 部分的数据库编号相匹配。

op 必须是以下值之一：

- = 或 == (等式)
- !=、<> 或 << (不等式)
- <= 或 =< (小于或等于)
- >= 或 => (大于或等于)
- < (小于)
- > (大于)

注意

字符串值仅支持等式或不等式。

value 是在表达式中使用的值。字符串值需要用单引号或双引号括起来，但这些引号不是值的一部分。如果希望左引号或右引号成为字符串的一部分，则必须将其输入两次。注意，对于在字符串当中出现的单引号，则不能输入两次。

设置动态陷阱时，分别使用 **k**、**m**、**g** 或 **t** 指定单位千字节 (KB)、兆字节 (MB)、千兆字节 (GB) 或千吉字节 (TB)。例如，您可以使用以下语句设置当前高速缓存大小超过 200 MB 时要触发的动态陷阱：

```
Trap1=1.3.6.1.4.1.897.2.1.1.11.1 > 200M
```

您可以在 *sasnmplib.ini* 文件中指定任意多个陷阱字段。该陷阱使用的 OID 为 1.3.6.1.4.1.897.2.4.1，随陷阱发送的数据包括：

- 陷阱编号（从 1 开始，SQL Anywhere SNMP Agent 发送的第一个动态陷阱的编号为 1）
- 数据库索引
- 数据库名称陷阱索引（来自 *sasnmplib.ini* 文件）
- 变量名
- 变量值（这是变量的当前值，不一定是阈值）

动态陷阱行为

在触发一个动态陷阱后，只有导致触发该陷阱的条件更改为 FALSE 后又更改为 TRUE，才会再次发送该陷阱。

例如，如果使用 1.1.11.1 >= 51200K 设置了一个动态陷阱，则当服务器的高速缓存大小达到 50 MB (= 51200 KB) 时会触发该陷阱，同时该动态陷阱被禁用，因此不会再发送陷阱。重新启用该陷

阱的唯一方式是高速缓存大小后来下降到低于 50 MB。当高速缓存大小再次增至 50 MB 时，您将获得通知。

陷阱示例

陷阱信息	说明
Trap1=1.1.5 > 10000	从服务器发送的字节数超过 10000 时发送的陷阱。
Trap2=1.3.6.1.2.1.39.1.4.1.4.14.1 >= 10485760	事务日志文件的大小超过 10 MB 时发送的陷阱。

SQL Anywhere MIB 参考

SNMP 代理支持的对象标识符 (OID) (包括它们的名称、类型和其它信息) 的列表存储在一个名为管理信息库 (Management Information Base, 简称 MIB) 的文件中。以下各节列出了可使用 SQL Anywhere SNMP Extension Agent 检索和设置的统计信息、属性和选项。

另请参见

- “了解 SNMP” 一节第 957 页

Agent

Agent 表列出了有关 SQL Anywhere SNMP Extension Agent 的信息。

星号 (*) 标记的为可写属性。值 *n* 是 *sasnmp.ini* 文件中的数据库编号。

OID	类型	名称	返回值
1.3.6.1.4.1.897.2.3.1	String	saVersion	Agent 版本
1.3.6.1.4.1.897.2.3.2. <i>n</i>	String	saDBConnStr	连接字符串
1.3.6.1.4.1.897.2.3.3. <i>n</i>	Integer32	saConnected*	如果代理已连接, 则为 1, 否则为 0
1.3.6.1.4.1.897.2.3.4. <i>n</i>	Integer32	saStarted*	如果数据库已启动, 则为 1, 否则为 0
1.3.6.1.4.1.897.2.3.5. <i>n</i>	String	saProc*	" "
1.3.6.1.4.1.897.2.3.6	String	saRestart*	0

saMetaData 表

SQL Anywhere MIB 中包含以下元数据表:

- saSrvMetaData.saSrvStatMetaDataTable
- saSrvMetaData.saSrvPropMetaDataTable
- saSrvMetaData.saDbStatMetaDataTable
- saSrvMetaData.saDbPropMetaDataTable
- saSrvMetaData.saDbOptMetaDataTable

saSrvMetaData.saSrvStatMetaDataTable

此表包含有关数据库服务器统计信息的元数据。

值 *db* 是 *sasnmplib.ini* 文件中的数据库编号。

OID	类型	名称	返回值
1.3.6.1.4.1.897.2.4.1.1.1.1. <i>db</i>	Integer32	saSrvStatIndex	<i>db</i>
1.3.6.1.4.1.897.2.4.1.1.1.2. <i>db</i>	Integer32	saSrvStatObjType	1 ¹
1.3.6.1.4.1.897.2.4.1.1.1.3. <i>db</i>	Integer32	saSrvStatType	1 ²
1.3.6.1.4.1.897.2.4.1.1.1.4. <i>db</i>	OID	saSrvStatOID	SQL Anywhere MIB 条目的 OID ³
1.3.6.1.4.1.897.2.4.1.1.1.5. <i>db</i>	String	saSrvStatName	统计信息名称

¹ 值: 1 为服务器, 2 为数据库

² 值: 1 为统计信息, 2 为属性, 3 为选项

³ 返回的 OID 不包括数据库编号。必须先向该 OID 添加数据库编号, 才能在查询中使用它。

saSrvMetaData.saSrvPropMetaDataTable

此表包含有关数据库服务器属性的元数据。

值 *db* 是 *sasnmplib.ini* 文件中的数据库编号。

OID	类型	名称	返回值
1.3.6.1.4.1.897.2.4.1.2.1.1. <i>db</i>	Integer32	saSrvPropIndex	<i>db</i>
1.3.6.1.4.1.897.2.4.1.2.1.2. <i>db</i>	Integer32	saSrvPropObjType	1 ¹
1.3.6.1.4.1.897.2.4.1.2.1.3. <i>db</i>	Integer32	saSrvPropType	2 ²
1.3.6.1.4.1.897.2.4.1.2.1.4. <i>db</i>	OID	saSrvPropOID	SQL Anywhere MIB 条目的 OID ³
1.3.6.1.4.1.897.2.4.1.2.1.5. <i>db</i>	String	saSrvPropName	属性名称

¹ 值: 1 为服务器, 2 为数据库

² 值: 1 为统计信息, 2 为属性, 3 为选项

³ 返回的 OID 不包括数据库编号。必须先向该 OID 添加数据库编号, 才能在查询中使用它。

saDbMetaData.saDbStatMetaDataTable

此表包含有关数据库统计信息的元数据。

值 *db* 是 *sasmp.ini* 文件中的数据库编号。

OID	类型	名称	返回值
1.3.6.1.4.1.897.2.4.2.1.1.1. <i>db</i>	Integer32	saDbStatIndex	<i>db</i>
1.3.6.1.4.1.897.2.4.2.1.1.2. <i>db</i>	Integer32	saDbStatObjType	2 ¹
1.3.6.1.4.1.897.2.4.2.1.1.3. <i>db</i>	Integer32	saDbStatType	1 ²
1.3.6.1.4.1.897.2.4.2.1.1.4. <i>db</i>	OID	saDbStatOID	SQL Anywhere MIB 条目的 OID ³
1.3.6.1.4.1.897.2.4.2.1.1.5. <i>db</i>	String	saDbStatName	统计信息名称

¹ 值: 1 为服务器, 2 为数据库

² 值: 1 为统计信息, 2 为属性, 3 为选项

³ 返回的 OID 不包括数据库编号。必须先向该 OID 添加数据库编号, 才能在查询中使用它。

saDbMetaData.saDbPropMetaDataTable

此表包含有关数据库属性的元数据。

值 *db* 是 *sasmp.ini* 文件中的数据库编号。

OID	类型	名称	返回值
1.3.6.1.4.1.897.2.4.2.2.1.1. <i>db</i>	Integer32	saDbPropIndex	<i>db</i>
1.3.6.1.4.1.897.2.4.2.2.1.2. <i>db</i>	Integer32	saDbPropObjType	2 ¹
1.3.6.1.4.1.897.2.4.2.2.1.3. <i>db</i>	Integer32	saDbPropType	2 ²
1.3.6.1.4.1.897.2.4.2.2.1.4. <i>db</i>	OID	saDbPropOID	SQL Anywhere MIB 条目的 OID ³
1.3.6.1.4.1.897.2.4.2.2.1.5. <i>db</i>	String	saDbPropName	属性名称

¹ 值: 1 为服务器, 2 为数据库

² 值: 1 为统计信息, 2 为属性, 3 为选项

³ 返回的 OID 不包括数据库编号。必须先向该 OID 添加数据库编号, 才能在查询中使用它。

saDbMetaData.saDbOptMetaDataTable

此表包含有关数据库选项的元数据。

值 *db* 是 *sasnmplib.ini* 文件中的数据库编号。

OID	类型	名称	返回值
1.3.6.1.4.1.897.2.4.2.1.1.1. <i>db</i>	Integer32	saDbOptIndex	<i>db</i>
1.3.6.1.4.1.897.2.4.2.1.1.2. <i>db</i>	Integer32	saDbOptObjType	2 ¹
1.3.6.1.4.1.897.2.4.2.1.1.3. <i>db</i>	Integer32	saDbOptType	3 ²
1.3.6.1.4.1.897.2.4.2.1.1.4. <i>db</i>	OID	saDbOptOID	SQL Anywhere MIB 条目的 OID ³
1.3.6.1.4.1.897.2.4.2.1.1.5. <i>db</i>	String	saDbOptName	选项名称

¹ 值: 1 为服务器, 2 为数据库

² 值: 1 为统计信息, 2 为属性, 3 为选项

³ 返回的 OID 不包括数据库编号。必须先向该 OID 添加数据库编号, 才能在查询中使用它。

SQL Anywhere MIB 服务器统计信息

此表列出了可使用 SQL Anywhere SNMP Extension Agent 检索的数据库服务器统计信息的 OID 和名称。

值 *n* 是 *sasnmplib.ini* 文件中的数据库编号。

有关数据库服务器统计信息的详细信息, 请参见“数据库服务器属性”一节第 588 页和“数据库属性”一节第 601 页。

OID	类型	名称	统计信息
1.3.6.1.4.1.897.2.1.1. <i>n</i>	Integer32	srvStatActiveReq	ActiveReq
1.3.6.1.4.1.897.2.1.1. <i>n</i>	Integer32	srvStatAvailIO	AvailIO
1.3.6.1.4.1.897.2.1.1. <i>n</i>	Counter64	srvStatBytesReceived	BytesReceived
1.3.6.1.4.1.897.2.1.1. <i>n</i>	Counter64	srvStatBytesReceivedUncomp	BytesReceivedUncomp
1.3.6.1.4.1.897.2.1.1. <i>n</i>	Counter64	srvStatBytesSent	BytesSent
1.3.6.1.4.1.897.2.1.1. <i>n</i>	Counter64	srvStatBytesSentUncomp	BytesSentUncomp
1.3.6.1.4.1.897.2.1.1. <i>n</i>	Counter64	srvStatCacheHits	CacheHits
1.3.6.1.4.1.897.2.1.1. <i>n</i>	Integer32	srvStatCachePinned	CachePinned

OID	类型	名称	统计信息
1.3.6.1.4.1.897.2.1.1.n	Counter64	srvStatCacheRead	CacheRead
1.3.6.1.4.1.897.2.1.1.n	Counter64	srvStatCacheReplacements	CacheReplacements
1.3.6.1.4.1.897.2.1.1.n	Integer32	srvStatCurrentCacheSize	CurrentCacheSize
1.3.6.1.4.1.897.2.1.1.n	Counter64	srvStatDiskRead	DiskRead
1.3.6.1.4.1.897.2.1.1.n	Integer32	srvStatFreeBuffers	FreeBuffers
1.3.6.1.4.1.897.2.1.1.n	Integer32	srvStatInternal	Internal
1.3.6.1.4.1.897.2.1.1.n	Integer32	srvStatUniqueClientAddresses	UniqueClientAddresses
1.3.6.1.4.1.897.2.1.1.n	Integer32	srvStatLockedHeapPages	LockedHeapPages
1.3.6.1.4.1.897.2.1.1.n	Counter64	srvStatMainHeapBytes	MainHeapBytes
1.3.6.1.4.1.897.2.1.1.n	Integer32	srvStatMainHeapPages	MainHeapPages
1.3.6.1.4.1.897.2.1.1.n	Integer32	srvStatMapPhysicalMemoryEng	MapPhysicalMemoryEng
1.3.6.1.4.1.897.2.1.1.n	Integer32	srvStatMaxCacheSize	MaxCacheSize
1.3.6.1.4.1.897.2.1.1.n	Integer32	srvStatMinCacheSize	MinCacheSize
1.3.6.1.4.1.897.2.1.1.n	Counter64	srvStatMultiPacketsReceived	MultiPacketsReceived
1.3.6.1.4.1.897.2.1.1.n	Counter64	srvStatMultiPacketsSent	MultiPacketsSent
1.3.6.1.4.1.897.2.1.1.n	Counter64	srvStatPacketsReceived	PacketsReceived
1.3.6.1.4.1.897.2.1.1.n	Counter64	srvStatPacketsReceivedUncomp	PacketsReceivedUncomp
1.3.6.1.4.1.897.2.1.1.n	Counter64	srvStatPacketsSent	PacketsSent
1.3.6.1.4.1.897.2.1.1.n	Counter64	srvStatPacketsSentUncomp	PacketsSentUncomp
1.3.6.1.4.1.897.2.1.1.n	Integer32	srvStatPeakCacheSize	PeakCacheSize
1.3.6.1.4.1.897.2.1.1.n	Integer32	srvStatRemoteputWait	RemoteputWait
1.3.6.1.4.1.897.2.1.1.n	Integer32	srvStatReq	Req
1.3.6.1.4.1.897.2.1.1.n	Counter64	srvStatSendFail	SendFail

OID	类型	名称	统计信息
1.3.6.1.4.1.897.2.1.1.n	Integer32	srvStatTotalBuffers	TotalBuffers
1.3.6.1.4.1.897.2.1.1.n	Integer32	srvStatUnschReq	UnschReq
1.3.6.1.4.1.897.2.1.1.n	Integer32	srvStatInternal	Internal
1.3.6.1.4.1.897.2.1.1.n	Integer32	srvStatCacheFile	CacheFile
1.3.6.1.4.1.897.2.1.1.n	Integer32	srvStatCacheFileDirty	CacheFileDirty
1.3.6.1.4.1.897.2.1.1.n	Integer32	srvStatCacheAllocated	CacheAllocated
1.3.6.1.4.1.897.2.1.1.n	Integer32	srvStatCachePanics	CachePanics
1.3.6.1.4.1.897.2.1.1.n	Integer32	srvStatCacheFree	CacheFree
1.3.6.1.4.1.897.2.1.1.n	Integer32	srvStatCacheScavenges	CacheScavenges
1.3.6.1.4.1.897.2.1.1.n	Integer32	srvStatCacheScavengeVisited	CacheScavengeVisited
1.3.6.1.4.1.897.2.1.1.n	Integer32	srvStatLockedCursorPages	LockedCursorPages
1.3.6.1.4.1.897.2.1.1.n	Integer32	srvStatQueryHeapPages	QueryHeapPages
1.3.6.1.4.1.897.2.1.1.n	Integer32	srvStatCarverHeapPages	CarverHeapPages
1.3.6.1.4.1.897.2.1.1.n	Integer32	srvStatHeapsRelocatable	HeapsRelocatable
1.3.6.1.4.1.897.2.1.1.n	Integer32	srvStatHeapsLocked	HeapsLocked
1.3.6.1.4.1.897.2.1.1.n	Integer32	srvStatHeapsQuery	HeapsQuery
1.3.6.1.4.1.897.2.1.1.n	Integer32	srvStatHeapsCarver	HeapsCarver
1.3.6.1.4.1.897.2.1.1.n	Integer32	srvStatMultiPageAllocs	MultiPageAllocs
1.3.6.1.4.1.897.2.1.1.n	Integer32	srvStatRequestsReceived	RequestsReceived
1.3.6.1.4.1.897.2.1.1.n	Integer32	srvStatExchangeTasks	ExchangeTasks
1.3.6.1.4.1.897.2.1.1.n	Integer32	srvStatClientStmtCacheHits	ClientStmtCacheHits
1.3.6.1.4.1.897.2.1.1.n	Integer32	srvStatClientStmtCacheMisses	ClientStmtCacheMisses
1.3.6.1.4.1.897.2.1.1.n	Integer32	srvStatQueryMemActiveCurr	QueryMemActiveCurr
1.3.6.1.4.1.897.2.1.1.n	Integer32	srvStatQueryMemActiveEst	QueryMemActiveEst

OID	类型	名称	统计信息
1.3.6.1.4.1.897.2.1.1.n	Integer32	srvStatQueryMemGrantWaiting	QueryMemGrantWaiting
1.3.6.1.4.1.897.2.1.1.n	Integer32	srvStatQueryMemGrantRequested	QueryMemGrantRequested
1.3.6.1.4.1.897.2.1.1.n	Integer32	srvStatQueryMemGrantWaited	QueryMemGrantWaited
1.3.6.1.4.1.897.2.1.1.n	Integer32	srvStatQueryMemGrantFailed	QueryMemGrantFailed
1.3.6.1.4.1.897.2.1.1.n	Integer32	srvStatQueryMemGrantGranted	QueryMemGrantGranted
1.3.6.1.4.1.897.2.1.1.n	Integer32	srvStatQueryMemExtraAvail	QueryMemExtraAvail

SQL Anywhere MIB 服务器属性

下表列出了可使用 SQL Anywhere SNMP Extension Agent 检索的数据库服务器属性的 OID 和名称。

星号 (*) 标记的为可写属性。值 *n* 是 *sasnmplib.ini* 文件中的数据库编号。

有关数据库服务器属性的详细信息，请参见“数据库属性”一节第 601 页。

OID	类型	名称	属性
1.3.6.1.4.1.897.2.1.2.1.n	String	srvPropInternal	Internal
1.3.6.1.4.1.897.2.1.2.2.n	String	srvPropCharSet	CharSet
1.3.6.1.4.1.897.2.1.2.3.n	String	srvPropCommandLine	CommandLine
1.3.6.1.4.1.897.2.1.2.4.n	String	srvPropCompactPlatformVer	CompactPlatformVer
1.3.6.1.4.1.897.2.1.2.5.n	String	srvPropCompanyName	CompanyName
1.3.6.1.4.1.897.2.1.2.6.n	String	srvPropConnsDisabled*	ConnsDisabled
1.3.6.1.4.1.897.2.1.2.7.n	String	srvPropConsoleLogFile	ConsoleLogFile
1.3.6.1.4.1.897.2.1.2.8.n	String	srvPropDefaultCollation	DefaultCollation
1.3.6.1.4.1.897.2.1.2.9.n	String	srvPropIdleTimeout	IdleTimeout
1.3.6.1.4.1.897.2.1.2.10.n	String	srvPropIsIQ	IsIQ

OID	类型	名称	属性
1.3.6.1.4.1.897.2.1.2.11..n	String	srvPropInternal	Internal
1.3.6.1.4.1.897.2.1.2.12..n	String	srvPropIsNetworkServer	IsNetworkServer
1.3.6.1.4.1.897.2.1.2.13..n	String	srvPropIsRuntimeServer	IsRuntimeServer
1.3.6.1.4.1.897.2.1.2.14..n	String	srvPropInternal	Internal
1.3.6.1.4.1.897.2.1.2.15..n	String	srvPropLanguage	Language
1.3.6.1.4.1.897.2.1.2.16..n	String	srvPropLegalCopyright	LegalCopyright
1.3.6.1.4.1.897.2.1.2.17..n	String	srvPropLegalTrademarks	LegalTrademarks
1.3.6.1.4.1.897.2.1.2.18..n	String	srvPropLicenseCount	LicenseCount
1.3.6.1.4.1.897.2.1.2.19..n	String	srvPropLicensedCompany	LicensedCompany
1.3.6.1.4.1.897.2.1.2.20..n	String	srvPropLicensedUser	LicensedUser
1.3.6.1.4.1.897.2.1.2.21..n	String	srvPropLicenseType	LicenseType
1.3.6.1.4.1.897.2.1.2.22..n	String	srvPropLivenessTimeout*	LivenessTimeout
1.3.6.1.4.1.897.2.1.2.23..n	String	srvPropMachineName	MachineName
1.3.6.1.4.1.897.2.1.2.24..n	String	srvPropMaxMessage	MaxMessage
1.3.6.1.4.1.897.2.1.2.25..n	String	srvPropMessageWindowSize	MessageWindowSize
1.3.6.1.4.1.897.2.1.2.26..n	String	srvPropName	名称
1.3.6.1.4.1.897.2.1.2.27..n	String	srvPropNativeProcessorArchitecture	NativeProcessorArchitecture
1.3.6.1.4.1.897.2.1.2.28..n	String	srvPropNumPhysicalProcessors	NumPhysicalProcessors
1.3.6.1.4.1.897.2.1.2.29..n	String	srvPropInternal	Internal
1.3.6.1.4.1.897.2.1.2.30..n	String	srvPropOmniIdentifier	OmniIdentifier
1.3.6.1.4.1.897.2.1.2.31..n	String	srvPropPageSize	PageSize
1.3.6.1.4.1.897.2.1.2.32..n	String	srvPropPlatform	Platform
1.3.6.1.4.1.897.2.1.2.33..n	String	srvPropPlatformVer	PlatformVer

OID	类型	名称	属性
1.3.6.1.4.1.897.2.1.2.34..n	String	srvPropProcessCPU	ProcessCPU
1.3.6.1.4.1.897.2.1.2.35..n	String	srvPropProcessCPUSystem	ProcessCPUSystem
1.3.6.1.4.1.897.2.1.2.36..n	String	srvPropProcessCPUUser	ProcessCPUUser
1.3.6.1.4.1.897.2.1.2.37..n	String	srvPropProcessorArchitecture	ProcessorArchitecture
1.3.6.1.4.1.897.2.1.2.38..n	String	srvPropProductName	ProductName
1.3.6.1.4.1.897.2.1.2.39..n	String	srvPropProductVersion	ProductVersion
1.3.6.1.4.1.897.2.1.2.40..n	String	srvPropQuittingTime*	QuittingTime
1.3.6.1.4.1.897.2.1.2.41..n	String	srvPropRememberLastStatement*	RememberLastStatement
1.3.6.1.4.1.897.2.1.2.42..n	String	srvPropRequestFilterConn	RequestFilterConn
1.3.6.1.4.1.897.2.1.2.43..n	String	srvPropRequestFilterDB	RequestFilterDB
1.3.6.1.4.1.897.2.1.2.44..n	String	srvPropRequestLogFile*	RequestLogFile
1.3.6.1.4.1.897.2.1.2.45..n	String	srvPropRequestLogging*	RequestLogging
1.3.6.1.4.1.897.2.1.2.46..n	String	srvPropRequestLogMaxSize	RequestLogMaxSize
1.3.6.1.4.1.897.2.1.2.47..n	String	srvPropStartTime	StartTime
1.3.6.1.4.1.897.2.1.2.48..n	String	srvPropTempDir	TempDir
1.3.6.1.4.1.897.2.1.2.49..n	String	srvPropMultiProgrammingLevel	MultiProgrammingLevel
1.3.6.1.4.1.897.2.1.2.50..n	String	srvPropTimeZoneAdjustment	TimeZoneAdjustment
1.3.6.1.4.1.897.2.1.2.51..n	String	srvPropHttpPorts	HttpPorts
1.3.6.1.4.1.897.2.1.2.52..n	String	srvPropHttpsPorts	HttpsPorts
1.3.6.1.4.1.897.2.1.2.53..n	String	srvPropProfileFilterConn	ProfileFilterConn
1.3.6.1.4.1.897.2.1.2.54..n	String	srvPropProfileFilterUser	ProfileFilterUser
1.3.6.1.4.1.897.2.1.2.55..n	String	srvPropRequestLogNumFiles	RequestLogNumFiles
1.3.6.1.4.1.897.2.1.2.56..n	String	srvPropIsFipsAvailable	IsFipsAvailable

OID	类型	名称	属性
1.3.6.1.4.1.897.2.1.2.57..n	String	srvPropFipsMode	FipsMode
1.3.6.1.4.1.897.2.1.2.58..n	String	srvPropStartDBPermission	StartDBPermission
1.3.6.1.4.1.897.2.1.2.59..n	String	srvPropServerName	ServerName
1.3.6.1.4.1.897.2.1.2.60..n	String	srvPropRememberLastPlan	RememberLastPlan
1.3.6.1.4.1.897.2.1.2.61..n	String	srvPropInternal	Internal
1.3.6.1.4.1.897.2.1.2.62..n	String	srvPropInternal	Internal
1.3.6.1.4.1.897.2.1.2.63..n	String	srvPropRequestTiming	RequestTiming
1.3.6.1.4.1.897.2.1.2.64..n	String	srvPropCacheSizingStatistics	CacheSizingStatistics
1.3.6.1.4.1.897.2.1.2.65..n	String	srvPropConsoleLogMaxSize	ConsoleLogMaxSize
1.3.6.1.4.1.897.2.1.2.66..n	String	srvPropDebuggingInformation	DebuggingInformation
1.3.6.1.4.1.897.2.1.2.67..n	String	srvPropMessage	Message
1.3.6.1.4.1.897.2.1.2.68..n	String	srvPropMessageText	MessageText
1.3.6.1.4.1.897.2.1.2.69..n	String	srvPropMessageTime	MessageTime
1.3.6.1.4.1.897.2.1.2.70..n	String	srvPropIsRsaAvailable	IsRsaAvailable
1.3.6.1.4.1.897.2.1.2.71..n	String	srvPropIsEccAvailable	IsEccAvailable
1.3.6.1.4.1.897.2.1.2.72..n	String	srvPropMaxConnections	MaxConnections
1.3.6.1.4.1.897.2.1.2.73..n	String	srvPropNumLogicalProcessors	NumLogicalProcessors
1.3.6.1.4.1.897.2.1.2.74..n	String	srvPropNumLogicalProcessorsUsed	NumLogicalProcessorsUsed
1.3.6.1.4.1.897.2.1.2.75..n	String	srvPropNumPhysicalProcessorsUsed	NumPhysicalProcessorsUsed
1.3.6.1.4.1.897.2.1.2.76..n	String	srvPropDefaultNcharCollation	DefaultNcharCollation
1.3.6.1.4.1.897.2.1.2.77..n	String	srvPropCollectStatistics	CollectStatistics

OID	类型	名称	属性
1.3.6.1.4.1.897.2.1.2.78..n	String	srvPropFirstOption	FirstOption
1.3.6.1.4.1.897.2.1.2.79..n	String	srvPropLastOption	LastOption
1.3.6.1.4.1.897.2.1.2.80..n	String	srvPropLastConnectionProperty	LastConnectionProperty
1.3.6.1.4.1.897.2.1.2.81..n	String	srvPropLastDatabaseProperty	LastDatabaseProperty
1.3.6.1.4.1.897.2.1.2.82..n	String	srvPropLastServerProperty	LastServerProperty
1.3.6.1.4.1.897.2.1.2.83..n	String	srvPropWebClientLogging	WebClientLogging
1.3.6.1.4.1.897.2.1.2.84..n	String	srvPropWebClientLogFile	WebClientLogFile
1.3.6.1.4.1.897.2.1.2.85..n	String	srvPropHttpNumConnections	HttpNumConnections
1.3.6.1.4.1.897.2.1.2.86..n	String	srvPropHttpsNumConnections	HttpsNumConnections
1.3.6.1.4.1.897.2.1.2.87..n	String	srvPropInternal	Internal
1.3.6.1.4.1.897.2.1.2.88..n	String	srvPropHttpNumActiveReq	HttpNumActiveReq
1.3.6.1.4.1.897.2.1.2.89..n	String	srvPropHttpsNumActiveReq	HttpsNumActiveReq
1.3.6.1.4.1.897.2.1.2.90..n	String	srvPropHttpNumSessions	HttpNumSessions
1.3.6.1.4.1.897.2.1.2.91..n	String	srvPropQueryMemPages	QueryMemPages
1.3.6.1.4.1.897.2.1.2.92..n	String	srvPropQueryMemGrantBase	QueryMemGrantBase
1.3.6.1.4.1.897.2.1.2.93..n	String	srvPropQueryMemGrantBaseMI	QueryMemGrantBaseMI
1.3.6.1.4.1.897.2.1.2.94..n	String	srvPropQueryMemGrantExtra	QueryMemGrantExtra
1.3.6.1.4.1.897.2.1.2.95..n	String	srvPropQueryMemActiveMax	QueryMemActiveMax
1.3.6.1.4.1.897.2.1.2.96..n	String	srvPropQueryMemPercentOfCache	QueryMemPercentOfCache
1.3.6.1.4.1.897.2.1.2.97..n	String	srvPropMessageCategoryLimit	MessageCategoryLimit

OID	类型	名称	属性
1.3.6.1.4.1.897.2.1.2.98..n	String	srvPropInternal	Internal
1.3.6.1.4.1.897.2.1.2.99..n	String	srvPropInternal	Internal
1.3.6.1.4.1.897.2.1.2.100..n	String	srvPropIsService	IsService
1.3.6.1.4.1.897.2.1.2.101..n	String	srvPropTcpIpAddresses	TcpIpAddresses
1.3.6.1.4.1.897.2.1.2.102..n	String	srvPropHttpAddresses	HttpAddresses
1.3.6.1.4.1.897.2.1.2.103..n	String	srvPropHttpsAddresses	HttpsAddresses
1.3.6.1.4.1.897.2.1.2.104..n	String	srvPropInternal	Internal
1.3.6.1.4.1.897.2.1.2.105..n	String	srvPropRemoteCapability	RemoteCapability
1.3.6.1.4.1.897.2.1.2.106..n	String	srvPropMaxRemoteCapabilit y	MaxRemoteCapability
1.3.6.1.4.1.897.2.1.2.107..n	String	srvPropEventTypeName	EventTypeName
1.3.6.1.4.1.897.2.1.2.108..n	String	srvPropEventTypeDef	EventTypeDef
1.3.6.1.4.1.897.2.1.2.109..n	String	srvPropMaxEventType	MaxEventType
1.3.6.1.4.1.897.2.1.2.110..n	String	srvPropInternal	Internal
1.3.6.1.4.1.897.2.1.2.111..n	String	srvPropInternal	Internal
1.3.6.1.4.1.897.2.1.2.112..n	String	srvPropServerEdition	ServerEdition

SQL Anywhere MIB 数据库统计信息

下表列出了可使用 SQL Anywhere SNMP Extension Agent 检索的数据库统计信息的 OID 和名称。

值 *n* 是 *sasmp.ini* 文件中的数据库编号。

有关数据库服务器统计信息的详细信息，请参见“数据库服务器属性”一节第 588 页和“数据库属性”一节第 601 页。

OID	类型	名称	统计信息
1.3.6.1.4.1.897.2.2.1.n	Counter64	dbStatCacheHits	CacheHits
1.3.6.1.4.1.897.2.2.1.n	Integer32	dbStatCacheReadIndInt	CacheReadIndInt

OID	类型	名称	统计信息
1.3.6.1.4.1.897.2.2.1.n	Integer32	dbStatCacheReadIndLeaf	CacheReadIndLeaf
1.3.6.1.4.1.897.2.2.1.n	Counter64	dbStatCacheRead	CacheRead
1.3.6.1.4.1.897.2.2.1.n	Integer32	dbStatCacheReadTable	CacheReadTable
1.3.6.1.4.1.897.2.2.1.n	Integer32	dbStatChkpt	Chkpt
1.3.6.1.4.1.897.2.2.1.n	Integer32	dbStatChkptFlush	ChkptFlush
1.3.6.1.4.1.897.2.2.1.n	Integer32	dbStatChkptPage	ChkptPage
1.3.6.1.4.1.897.2.2.1.n	Integer32	dbStatCheckpointUrgency	CheckpointUrgency
1.3.6.1.4.1.897.2.2.1.n	Integer32	dbStatCheckpointLogBitmapSize	CheckpointLogBitmapSize
1.3.6.1.4.1.897.2.2.1.n	Integer32	dbStatCheckpointLogBitmapPagesWritten	CheckpointLogBitmapPagesWritten
1.3.6.1.4.1.897.2.2.1.n	Integer32	dbStatCheckpointLogCommitToDisk	CheckpointLogCommitToDisk
1.3.6.1.4.1.897.2.2.1.n	Integer32	dbStatCheckpointLogPageInUse	CheckpointLogPageInUse
1.3.6.1.4.1.897.2.2.1.n	Integer32	dbStatCheckpointLogPagesRelocated	CheckpointLogPagesRelocated
1.3.6.1.4.1.897.2.2.1.n	Integer32	dbStatCheckpointLogSavePreimage	CheckpointLogSavePreimage
1.3.6.1.4.1.897.2.2.1.n	Integer32	dbStatCheckpointLogSize	CheckpointLogSize
1.3.6.1.4.1.897.2.2.1.n	Integer32	dbStatCheckpointLogWrites	CheckpointLogWrites
1.3.6.1.4.1.897.2.2.1.n	Integer32	dbStatCheckpointLogPagesWritten	CheckpointLogPagesWritten
1.3.6.1.4.1.897.2.2.1.n	Integer32	dbStatCommitFile	CommitFile
1.3.6.1.4.1.897.2.2.1.n	Integer32	dbStatConnCount	ConnCount
1.3.6.1.4.1.897.2.2.1.n	Integer32	dbStatCurrIO	CurrIO
1.3.6.1.4.1.897.2.2.1.n	Integer32	dbStatCurrRead	CurrRead

OID	类型	名称	统计信息
1.3.6.1.4.1.897.2.2.1.n	Integer32	dbStatCurrWrite	CurrWrite
1.3.6.1.4.1.897.2.2.1.n	Integer32	dbStatDiskReadIndInt	DiskReadIndInt
1.3.6.1.4.1.897.2.2.1.n	Integer32	dbStatDiskReadIndLeaf	DiskReadIndLeaf
1.3.6.1.4.1.897.2.2.1.n	Counter64	dbStatDiskRead	DiskRead
1.3.6.1.4.1.897.2.2.1.n	Integer32	dbStatDiskReadTable	DiskReadTable
1.3.6.1.4.1.897.2.2.1.n	Counter64	dbStatDiskWrite	DiskWrite
1.3.6.1.4.1.897.2.2.1.n	Integer32	dbStatExtendDB	ExtendDB
1.3.6.1.4.1.897.2.2.1.n	Integer32	dbStatExtendTempWrite	ExtendTempWrite
1.3.6.1.4.1.897.2.2.1.n	Integer32	dbStatFullCompare	FullCompare
1.3.6.1.4.1.897.2.2.1.n	Integer32	dbStatGetData	GetData
1.3.6.1.4.1.897.2.2.1.n	Integer32	dbStatIdleCheck	IdleCheck
1.3.6.1.4.1.897.2.2.1.n	Integer32	dbStatIdleChkpt	IdleChkpt
1.3.6.1.4.1.897.2.2.1.n	Integer32	dbStatIdleChkTime	IdleChkTime
1.3.6.1.4.1.897.2.2.1.n	Integer32	dbStatIdleWrite	IdleWrite
1.3.6.1.4.1.897.2.2.1.n	Integer32	dbStatIndAdd	IndAdd
1.3.6.1.4.1.897.2.2.1.n	Integer32	dbStatIndLookup	IndLookup
1.3.6.1.4.1.897.2.2.1.n	Integer32	dbStatIOToRecover	IOToRecover
1.3.6.1.4.1.897.2.2.1.n	Integer32	dbStatInternal	Internal
1.3.6.1.4.1.897.2.2.1.n	Integer32	dbStatInternal	Internal
1.3.6.1.4.1.897.2.2.1.n	Integer32	dbStatLockTablePages	LockTablePages
1.3.6.1.4.1.897.2.2.1.n	Integer32	dbStatInternal	Internal
1.3.6.1.4.1.897.2.2.1.n	Integer32	dbStatMaxIO	MaxIO
1.3.6.1.4.1.897.2.2.1.n	Integer32	dbStatMaxRead	MaxRead
1.3.6.1.4.1.897.2.2.1.n	Integer32	dbStatMaxWrite	MaxWrite

OID	类型	名称	统计信息
1.3.6.1.4.1.897.2.2.1.n	Counter64	dbStatPageRelocations	PageRelocations
1.3.6.1.4.1.897.2.2.1.n	Integer32	dbStatProcedurePages	ProcedurePages
1.3.6.1.4.1.897.2.2.1.n	Integer32	dbStatQueryCachePages	QueryCachePages
1.3.6.1.4.1.897.2.2.1.n	Integer32	dbStatQueryLowMemoryStrategy	QueryLowMemoryStrategy
1.3.6.1.4.1.897.2.2.1.n	Counter64	dbStatQueryRowsMaterialized	QueryRowsMaterialized
1.3.6.1.4.1.897.2.2.1.n	Integer32	dbStatRecoveryUrgency	RecoveryUrgency
1.3.6.1.4.1.897.2.2.1.n	Integer32	dbStatLogFreeCommit	LogFreeCommit
1.3.6.1.4.1.897.2.2.1.n	Integer32	dbStatLogWrite	LogWrite
1.3.6.1.4.1.897.2.2.1.n	Integer32	dbStatRelocatableHeapPages	RelocatableHeapPages
1.3.6.1.4.1.897.2.2.1.n	Integer32	dbStatRollbackLogPages	RollbackLogPages
1.3.6.1.4.1.897.2.2.1.n	Integer32	dbStatTempTablePages	TempTablePages
1.3.6.1.4.1.897.2.2.1.n	Integer32	dbStatTriggerPages	TriggerPages
1.3.6.1.4.1.897.2.2.1.n	Integer32	dbStatViewPages	ViewPages
1.3.6.1.4.1.897.2.2.1.n	Integer32	dbStatVersionStorePages	VersionStorePages
1.3.6.1.4.1.897.2.2.1.n	Integer32	dbStatSnapshotCount	SnapshotCount
1.3.6.1.4.1.897.2.2.1.n	Integer32	dbStatLockCount	LockCount
1.3.6.1.4.1.897.2.2.1.n	Integer32	dbStatCacheReadWorkTable	CacheReadWorkTable
1.3.6.1.4.1.897.2.2.1.n	Integer32	dbStatDiskReadWorkTable	DiskReadWorkTable
1.3.6.1.4.1.897.2.2.1.n	Integer32	dbStatPrepares	Prepares

SQL Anywhere MIB 数据库属性

下表列出了可使用 SQL Anywhere SNMP Extension Agent 检索的数据库属性的 OID 和名称。

星号 (*) 标记的为可写属性。值 *n* 是 *sasnmplib.ini* 文件中的数据库编号。

有关数据库服务器属性的详细信息，请参见“数据库属性”一节第 601 页。

OID	类型	名称	属性
1.3.6.1.4.1.897.2.2.2.1..n	String	dbPropAlias	Alias
1.3.6.1.4.1.897.2.2.2.2..n	String	dbPropAuditingTypes	AuditingTypes
1.3.6.1.4.1.897.2.2.2.3..n	String	dbPropBlankPadding	BlankPadding
1.3.6.1.4.1.897.2.2.2.4..n	String	dbPropInternal	Internal
1.3.6.1.4.1.897.2.2.2.5..n	String	dbPropCapabilities	Capabilities
1.3.6.1.4.1.897.2.2.2.6..n	String	dbPropCaseSensitive	CaseSensitive
1.3.6.1.4.1.897.2.2.2.7..n	String	dbPropInternal	Internal
1.3.6.1.4.1.897.2.2.2.8..n	String	dbPropCharSet	CharSet
1.3.6.1.4.1.897.2.2.2.9..n	String	dbPropChecksum	Checksum
1.3.6.1.4.1.897.2.2.2.10..n	String	dbPropInternal	Internal
1.3.6.1.4.1.897.2.2.2.11..n	String	dbPropCollation	Collation
1.3.6.1.4.1.897.2.2.2.12..n	String	dbPropInternal	Internal
1.3.6.1.4.1.897.2.2.2.13..n	String	dbPropInternal	Internal
1.3.6.1.4.1.897.2.2.2.14..n	String	dbPropCurrentRedoPos	CurrentRedoPos
1.3.6.1.4.1.897.2.2.2.15..n	String	dbPropDBFileFragments	DBFileFragments
1.3.6.1.4.1.897.2.2.2.16..n	String	dbPropDriveType	DriveType
1.3.6.1.4.1.897.2.2.2.17..n	String	dbPropEncryption	Encryption
1.3.6.1.4.1.897.2.2.2.18..n	String	dbPropFile	File
1.3.6.1.4.1.897.2.2.2.19..n	String	dbPropFileSize	FileSize
1.3.6.1.4.1.897.2.2.2.20..n	String	dbPropInternal	Internal
1.3.6.1.4.1.897.2.2.2.21..n	String	dbPropInternal	Internal
1.3.6.1.4.1.897.2.2.2.22..n	String	dbPropFreePages	FreePages
1.3.6.1.4.1.897.2.2.2.23..n	String	dbPropGlobalDBID	GlobalDBID
1.3.6.1.4.1.897.2.2.2.24..n	String	dbPropInternal	Internal

OID	类型	名称	属性
1.3.6.1.4.1.897.2.2.2.25..n	String	dbPropInternal	Internal
1.3.6.1.4.1.897.2.2.2.26..n	String	dbPropInternal	Internal
1.3.6.1.4.1.897.2.2.2.27..n	String	dbPropIQStore	IQStore
1.3.6.1.4.1.897.2.2.2.28..n	String	dbPropInternal	Internal
1.3.6.1.4.1.897.2.2.2.29..n	String	dbPropLanguage	Language
1.3.6.1.4.1.897.2.2.2.30..n	String	dbPropInternal	Internal
1.3.6.1.4.1.897.2.2.2.31..n	String	dbPropLogFileFragments	LogFileFragments
1.3.6.1.4.1.897.2.2.2.32..n	String	dbPropLogMirrorName	LogMirrorName
1.3.6.1.4.1.897.2.2.2.33..n	String	dbPropLogName	LogName
1.3.6.1.4.1.897.2.2.2.34..n	String	dbPropLTMGeneration	LTMGeneration
1.3.6.1.4.1.897.2.2.2.35..n	String	dbPropLTMTrunc	LTMTrunc
1.3.6.1.4.1.897.2.2.2.36..n	String	dbPropMultiByteCharSet	MultiByteCharSet
1.3.6.1.4.1.897.2.2.2.37..n	String	dbPropName	名称
1.3.6.1.4.1.897.2.2.2.38..n	String	dbPropInternal	Internal
1.3.6.1.4.1.897.2.2.2.39..n	String	dbPropPageSize	PageSize
1.3.6.1.4.1.897.2.2.2.40..n	String	dbPropInternal	Internal
1.3.6.1.4.1.897.2.2.2.41..n	String	dbPropProcedureProfiling*	ProcedureProfiling
1.3.6.1.4.1.897.2.2.2.42..n	String	dbPropReadOnly	ReadOnly
1.3.6.1.4.1.897.2.2.2.43..n	String	dbPropRemoteTrunc	RemoteTrunc
1.3.6.1.4.1.897.2.2.2.44..n	String	dbPropInternal	Internal
1.3.6.1.4.1.897.2.2.2.45..n	String	dbPropInternal	Internal
1.3.6.1.4.1.897.2.2.2.46..n	String	dbPropInternal	Internal
1.3.6.1.4.1.897.2.2.2.47..n	String	dbPropSyncTrunc	SyncTrunc
1.3.6.1.4.1.897.2.2.2.48..n	String	dbPropInternal	Internal

OID	类型	名称	属性
1.3.6.1.4.1.897.2.2.2.49..n	String	dbPropInternal	Internal
1.3.6.1.4.1.897.2.2.2.50..n	String	dbPropTempFileName	TempFileName
1.3.6.1.4.1.897.2.2.2.51..n	String	dbPropInternal	Internal
1.3.6.1.4.1.897.2.2.2.52..n	String	dbPropInternal	Internal
1.3.6.1.4.1.897.2.2.2.53..n	String	dbPropInternal	Internal
1.3.6.1.4.1.897.2.2.2.54..n	String	dbPropNextScheduleTime	NextScheduleTime
1.3.6.1.4.1.897.2.2.2.55..n	String	dbPropIdentitySignature	IdentitySignature
1.3.6.1.4.1.897.2.2.2.56..n	String	dbPropInternal	Internal
1.3.6.1.4.1.897.2.2.2.57..n	String	dbPropSnapshotIsolationState	SnapshotIsolationState
1.3.6.1.4.1.897.2.2.2.58..n	String	dbPropConnsDisabled	ConnsDisabled
1.3.6.1.4.1.897.2.2.2.59..n	String	dbPropPartnerState	PartnerState
1.3.6.1.4.1.897.2.2.2.60..n	String	dbPropArbiterState	ArbiterState
1.3.6.1.4.1.897.2.2.2.61..n	String	dbPropMirrorState	MirrorState
1.3.6.1.4.1.897.2.2.2.62..n	String	dbPropAlternateServerName	AlternateServerName
1.3.6.1.4.1.897.2.2.2.63..n	String	dbPropEncryptionScope	EncryptionScope
1.3.6.1.4.1.897.2.2.2.64..n	String	dbPropNcharCharSet	NcharCharSet
1.3.6.1.4.1.897.2.2.2.65..n	String	dbPropNcharCollation	NcharCollation
1.3.6.1.4.1.897.2.2.2.66..n	String	dbPropAccentSensitive	AccentSensitive
1.3.6.1.4.1.897.2.2.2.67..n	String	dbPropSendingTracingTo	SendingTracingTo
1.3.6.1.4.1.897.2.2.2.68..n	String	dbPropReceivingTracingFrom	ReceivingTracingFrom
1.3.6.1.4.1.897.2.2.2.69..n	String	dbPropIOParallelism	IOParallelism
1.3.6.1.4.1.897.2.2.2.70..n	String	dbPropJavaVM	JavaVM
1.3.6.1.4.1.897.2.2.2.71..n	String	dbPropDatabaseCleaner	DatabaseCleaner
1.3.6.1.4.1.897.2.2.2.72..n	String	dbPropHasCollationTailoring	HasCollationTailoring

OID	类型	名称	属性
1.3.6.1.4.1.897.2.2.2.73..n	String	dbPropCatalogCollation	CatalogCollation
1.3.6.1.4.1.897.2.2.2.74..n	String	dbPropHasEndianSwapFix	HasEndianSwapFix
1.3.6.1.4.1.897.2.2.2.75..n	String	dbPropAlternateMirrorServerName	AlternateMirrorServerName
1.3.6.1.4.1.897.2.2.2.76..n	String	dbPropOptionWatchList	OptionWatchList
1.3.6.1.4.1.897.2.2.2.77..n	String	dbPropOptionWatchAction	OptionWatchAction
1.3.6.1.4.1.897.2.2.2.78..n	String	dbPropMirrorMode	MirrorMode
1.3.6.1.4.1.897.2.2.2.79..n	String	dbPropHasNCHARLegacyCollationFix	HasNCHARLegacyCollationFix
1.3.6.1.4.1.897.2.2.2.80..n	String	dbPropInternal	Internal
1.3.6.1.4.1.897.2.2.2.81..n	String	dbPropAuthenticated	Authenticated

SQL Anywhere MIB 数据库选项

下表列出了可使用 SQL Anywhere SNMP Extension Agent 检索的数据库选项的 OID 和名称。

星号 (*) 标记的为可写选项。值 *n* 是 *sasnmp.ini* 文件中的数据库编号。

有关数据库属性的详细信息，请参见“按字母顺序排列的选项列表”一节第 477 页。

OID	类型	名称	选项
1.3.6.1.4.1.897.2.2.3.1..n	String	dbOptAllowNullsByDefault*	allow_nulls_by_default
1.3.6.1.4.1.897.2.2.3.2..n	String	dbOptAnsiNull*	ansinull
1.3.6.1.4.1.897.2.2.3.3..n	String	dbOptAnsiBlanks*	ansi_blanks
1.3.6.1.4.1.897.2.2.3.4..n	String	dbOptAnsiCloseCursorsOnRollback*	ansi_close_cursors_on_rollback
1.3.6.1.4.1.897.2.2.3.5..n	Integer32	dbOptInternal	Internal
1.3.6.1.4.1.897.2.2.3.6..n	String	dbOptAnsiPermissions*	ansi_permissions
1.3.6.1.4.1.897.2.2.3.7..n	String	dbOptAnsiUpdateConstraints*	ansi_update_constraints

OID	类型	名称	选项
1.3.6.1.4.1.897.2.2.3.8..n	String	dbOptAuditing*	auditing
1.3.6.1.4.1.897.2.2.3.9..n	String	dbOptAuditingOptions*	auditing_options
1.3.6.1.4.1.897.2.2.3.10..n	Integer32	dbOptInternal	Internal
1.3.6.1.4.1.897.2.2.3.11..n	String	dbOptBackgroundPriority*	background_priority
1.3.6.1.4.1.897.2.2.3.12..n	String	dbOptBlocking*	blocking
1.3.6.1.4.1.897.2.2.3.13..n	Integer32	dbOptBlockingTimeout*	blocking_timeout
1.3.6.1.4.1.897.2.2.3.14..n	String	dbOptChained*	chained
1.3.6.1.4.1.897.2.2.3.15..n	Integer32	dbOptCheckpointTime*	checkpoint_time
1.3.6.1.4.1.897.2.2.3.16..n	Integer32	dbOptCisOption*	cis_option
1.3.6.1.4.1.897.2.2.3.17..n	Integer32	dbOptCisRowsetSize*	cis_rowset_size
1.3.6.1.4.1.897.2.2.3.18..n	String	dbOptCloseOnEndtrans*	close_on_endtrans
1.3.6.1.4.1.897.2.2.3.19..n	String	dbOptConnectionAuthentication*	connection_authentication
1.3.6.1.4.1.897.2.2.3.20..n	String	dbOptContinueAfterRaiserror*	continue_after_raiserror
1.3.6.1.4.1.897.2.2.3.21..n	String	dbOptConversionError*	conversion_error
1.3.6.1.4.1.897.2.2.3.22..n	String	dbOptCooperativeCommits*	cooperative_commits
1.3.6.1.4.1.897.2.2.3.23..n	Integer32	dbOptCooperativeCommitTimeout*	cooperative_commit_timeout
1.3.6.1.4.1.897.2.2.3.24..n	String	dbOptDatabaseAuthentication*	database_authentication
1.3.6.1.4.1.897.2.2.3.25..n	String	dbOptDateFormat*	date_format
1.3.6.1.4.1.897.2.2.3.26..n	String	dbOptDateOrder*	date_order
1.3.6.1.4.1.897.2.2.3.27..n	String	dbOptDebugMessages*	debug_messages
1.3.6.1.4.1.897.2.2.3.28..n	String	dbOptDedicatedTask*	dedicated_task

OID	类型	名称	选项
1.3.6.1.4.1.897.2.2.3.29..n	Integer32	dbOptDefaultTimestampIncrement*	default_timestamp_increment
1.3.6.1.4.1.897.2.2.3.30..n	String	dbOptDelayedCommits*	delayed_commits
1.3.6.1.4.1.897.2.2.3.31..n	Integer32	dbOptDelayedCommitTimeout*	delayed_commit_timeout
1.3.6.1.4.1.897.2.2.3.32..n	Integer32	dbOptInternal	Internal
1.3.6.1.4.1.897.2.2.3.33..n	String	dbOptEscapeCharacter*	escape_character
1.3.6.1.4.1.897.2.2.3.34..n	String	dbOptExcludeOperators*	exclude_operators
1.3.6.1.4.1.897.2.2.3.35..n	String	dbOptExtendedJoinSyntax*	extended_join_syntax
1.3.6.1.4.1.897.2.2.3.36..n	String	dbOptFireTriggers*	fire_triggers
1.3.6.1.4.1.897.2.2.3.37..n	Integer32	dbOptFirstDayOfWeek*	first_day_of_week
1.3.6.1.4.1.897.2.2.3.38..n	Integer32	dbOptInternal	Internal
1.3.6.1.4.1.897.2.2.3.39..n	String	dbOptForceViewCreation*	force_view_creation
1.3.6.1.4.1.897.2.2.3.40..n	String	dbOptForXmlNullTreatment*	for_xml_null_treatment
1.3.6.1.4.1.897.2.2.3.41..n	Integer32	dbOptGlobalDatabaseId*	global_database_id
1.3.6.1.4.1.897.2.2.3.42..n	Integer32	dbOptIsolationLevel*	isolation_level
1.3.6.1.4.1.897.2.2.3.43..n	Integer32	dbOptInternal	Internal
1.3.6.1.4.1.897.2.2.3.44..n	Integer32	dbOptInternal	Internal
1.3.6.1.4.1.897.2.2.3.45..n	Integer32	dbOptInternal	Internal
1.3.6.1.4.1.897.2.2.3.46..n	Integer32	dbOptInternal	Internal
1.3.6.1.4.1.897.2.2.3.47..n	String	dbOptLockRejectedRows*	lock_rejected_rows
1.3.6.1.4.1.897.2.2.3.48..n	String	dbOptLoginMode*	login_mode
1.3.6.1.4.1.897.2.2.3.49..n	String	dbOptLoginProcedure*	login_procedure
1.3.6.1.4.1.897.2.2.3.50..n	Integer32	dbOptInternal	Internal
1.3.6.1.4.1.897.2.2.3.51..n	Integer32	dbOptInternal	Internal

OID	类型	名称	选项
1.3.6.1.4.1.897.2.2.3.52..n	Integer32	dbOptMaxCursorCount*	max_cursor_count
1.3.6.1.4.1.897.2.2.3.53..n	Integer32	dbOptMaxHashSize*	max_hash_size
1.3.6.1.4.1.897.2.2.3.54..n	Integer32	dbOptMaxPlansCached*	max_plans_cached
1.3.6.1.4.1.897.2.2.3.55..n	Integer32	dbOptMaxRecursiveIterations*	max_recursive_iterations
1.3.6.1.4.1.897.2.2.3.56..n	Integer32	dbOptMaxStatementCount*	max_statement_count
1.3.6.1.4.1.897.2.2.3.57..n	Integer32	dbOptInternal	Internal
1.3.6.1.4.1.897.2.2.3.58..n	Integer32	dbOptMinPasswordLength*	min_password_length
1.3.6.1.4.1.897.2.2.3.59..n	Integer32	dbOptInternal	Internal
1.3.6.1.4.1.897.2.2.3.60..n	Integer32	dbOptNearestCentury*	nearest_century
1.3.6.1.4.1.897.2.2.3.61..n	String	dbOptNonKeywords*	non_keywords
1.3.6.1.4.1.897.2.2.3.62..n	String	dbOptODBCDistinguishCharAndVarchar*	odbc_distinguish_char_and_varchar
1.3.6.1.4.1.897.2.2.3.63..n	String	dbOptOnCharsetConversionFailure*	on_charset_conversion_failure
1.3.6.1.4.1.897.2.2.3.64..n	String	dbOptOnTsqlError*	on_tsql_error
1.3.6.1.4.1.897.2.2.3.65..n	Integer32	dbOptInternal	Internal
1.3.6.1.4.1.897.2.2.3.66..n	String	dbOptOptimizationGoal*	optimization_goal
1.3.6.1.4.1.897.2.2.3.67..n	Integer32	dbOptOptimizationLevel*	optimization_level
1.3.6.1.4.1.897.2.2.3.68..n	Integer32	dbOptInternal	Internal
1.3.6.1.4.1.897.2.2.3.69..n	String	dbOptOptimizationWorkload*	optimization_workload
1.3.6.1.4.1.897.2.2.3.70..n	Integer32	dbOptPinnedCursorPercentOfCache*	pinned_cursor_percent_of_cache
1.3.6.1.4.1.897.2.2.3.71..n	Integer32	dbOptPrecision*	precision
1.3.6.1.4.1.897.2.2.3.72..n	String	dbOptPrefetch*	prefetch

OID	类型	名称	选项
1.3.6.1.4.1.897.2.2.3.73..n	String	dbOptPreserveSourceFormat*	preserve_source_format
1.3.6.1.4.1.897.2.2.3.74..n	String	dbOptPreventArticlePkeyUpdate*	prevent_article_pkey_update
1.3.6.1.4.1.897.2.2.3.75..n	Integer32	dbOptInternal	Internal
1.3.6.1.4.1.897.2.2.3.76..n	String	dbOptQuotedIdentifier*	quoted_identifier
1.3.6.1.4.1.897.2.2.3.77..n	String	dbOptReadPastDeleted*	read_past_deleted
1.3.6.1.4.1.897.2.2.3.78..n	Integer32	dbOptRecoveryTime*	recovery_time
1.3.6.1.4.1.897.2.2.3.79..n	String	dbOptReplicateAll*	replicate_all
1.3.6.1.4.1.897.2.2.3.80..n	String	dbOptReturnDateTimeAsString*	return_date_time_as_string
1.3.6.1.4.1.897.2.2.3.81..n	Integer32	dbOptInternal	Internal
1.3.6.1.4.1.897.2.2.3.82..n	Integer32	dbOptInternal	Internal
1.3.6.1.4.1.897.2.2.3.83..n	String	dbOptRowCounts*	row_counts
1.3.6.1.4.1.897.2.2.3.84..n	Integer32	dbOptScale*	scale
1.3.6.1.4.1.897.2.2.3.85..n	String	dbOptSortCollation*	sort_collation
1.3.6.1.4.1.897.2.2.3.86..n	String	dbOptSQLFlaggerErrorLevel*	sql_flagger_error_level
1.3.6.1.4.1.897.2.2.3.87..n	String	dbOptSQLFlaggerWarningLevel*	sql_flagger_warning_level
1.3.6.1.4.1.897.2.2.3.88..n	String	dbOptStringRtruncation*	string_rtruncation
1.3.6.1.4.1.897.2.2.3.89..n	String	dbOptSubsumeRowLocks*	subsume_row_locks
1.3.6.1.4.1.897.2.2.3.90..n	String	dbOptSuppressTDSDebugging*	suppress_tds_debugging
1.3.6.1.4.1.897.2.2.3.91..n	String	dbOptTDSEmptyStringIsNull*	tds_empty_string_is_null
1.3.6.1.4.1.897.2.2.3.92..n	Integer32	dbOptInternal	Internal
1.3.6.1.4.1.897.2.2.3.93..n	Integer32	dbOptInternal	Internal

OID	类型	名称	选项
1.3.6.1.4.1.897.2.2.3.94..n	Integer32	dbOptInternal	Internal
1.3.6.1.4.1.897.2.2.3.95..n	String	dbOptTimestampFormat*	timestamp_format
1.3.6.1.4.1.897.2.2.3.96..n	String	dbOptTimeFormat*	time_format
1.3.6.1.4.1.897.2.2.3.97..n	Integer32	dbOptTimeZoneAdjustment*	time_zone_adjustment
1.3.6.1.4.1.897.2.2.3.98..n	Integer32	dbOptInternal	Internal
1.3.6.1.4.1.897.2.2.3.99..n	String	dbOptTruncateTimestampValues*	truncate_timestamp_values
1.3.6.1.4.1.897.2.2.3.100..n	Integer32	dbOptInternal	Internal
1.3.6.1.4.1.897.2.2.3.101..n	Integer32	dbOptInternal	Internal
1.3.6.1.4.1.897.2.2.3.102..n	String	dbOptTsqlVariables*	tsql_variables
1.3.6.1.4.1.897.2.2.3.103..n	String	dbOptUpdateStatistics*	update_statistics
1.3.6.1.4.1.897.2.2.3.104..n	String	dbOptUpgradeDatabaseCapability*	upgrade_database_capability
1.3.6.1.4.1.897.2.2.3.105..n	String	dbOptUserEstimates*	user_estimates
1.3.6.1.4.1.897.2.2.3.106..n	String	dbOptWaitForCommit*	wait_for_commit
1.3.6.1.4.1.897.2.2.3.107..n	String	dbOptTempSpaceLimitCheck*	temp_space_limit_check
1.3.6.1.4.1.897.2.2.3.108..n	Integer32	dbOptRemoteIdleTimeout*	remote_idle_timeout
1.3.6.1.4.1.897.2.2.3.109..n	String	dbOptAnsiSubstring*	ansi_substring
1.3.6.1.4.1.897.2.2.3.110..n	String	dbOptOdbcDescribeBinaryAsVarbinary*	odbc_describe_binary_as_varbinary
1.3.6.1.4.1.897.2.2.3.111..n	String	dbOptRollbackOnDeadlock	rollback_on_deadlock
1.3.6.1.4.1.897.2.2.3.112..n	String	dbOptIntegratedServerName*	integrated_server_name
1.3.6.1.4.1.897.2.2.3.113..n	String	dbOptLogDeadlocks*	log_deadlocks
1.3.6.1.4.1.897.2.2.3.114..n	Integer32	dbOptInternal	Internal

OID	类型	名称	选项
1.3.6.1.4.1.897.2.2.3.115..n	String	dbOptWebserviceNamespaceHost*	webservice_namespace_host
1.3.6.1.4.1.897.2.2.3.116..n	Integer32	dbOptMaxQueryTasks*	max_query_tasks
1.3.6.1.4.1.897.2.2.3.117..n	Integer32	dbOptRequestTimeout*	request_timeout
1.3.6.1.4.1.897.2.2.3.118..n	String	dbOptSynchronizeMirrorOnCommit*	synchronize_mirror_on_commit
1.3.6.1.4.1.897.2.2.3.119..n	Integer32	dbOptHttpSessionTimeout*	http_session_timeout
1.3.6.1.4.1.897.2.2.3.120..n	Integer32	dbOptInternal	Internal
1.3.6.1.4.1.897.2.2.3.121..n	String	dbOptAllowSnapshotIsolation*	allow_snapshot_isolation
1.3.6.1.4.1.897.2.2.3.122..n	String	dbOptVerifyPasswordFunction*	verify_password_function
1.3.6.1.4.1.897.2.2.3.123..n	String	dbOptDefaultDbpace*	default_dbpace
1.3.6.1.4.1.897.2.2.3.124..n	String	dbOptCollectStatisticsOnDmlUpdates*	collect_statistics_on_dml_updates
1.3.6.1.4.1.897.2.2.3.125..n	String	dbOptJavaMainUserid*	java_main_userid
1.3.6.1.4.1.897.2.2.3.126..n	String	dbOptJavaLocation*	java_location
1.3.6.1.4.1.897.2.2.3.127..n	String	dbOptOemString*	oem_string
1.3.6.1.4.1.897.2.2.3.128..n	Integer32	dbOptMaxTempSpace*	max_temp_space
1.3.6.1.4.1.897.2.2.3.129..n	String	dbOptSecureFeatureKey*	secure_feature_key
1.3.6.1.4.1.897.2.2.3.130..n	String	dbOptMaterializedViewOptimization*	materialized_view_optimization
1.3.6.1.4.1.897.2.2.3.131..n	Integer32	dbOptUpdatableStatementIsolation*	updatable_statement_isolation
1.3.6.1.4.1.897.2.2.3.132..n	String	dbOptTsqlOuterJoins*	tsql_outer_joins
1.3.6.1.4.1.897.2.2.3.133..n	String	dbOptPostLoginProcedure*	post_login_procedure
1.3.6.1.4.1.897.2.2.3.134..n	String	dbOptConnAuditing*	conn_auditing

OID	类型	名称	选项
1.3.6.1.4.1.897.2.2.3.135.. <i>n</i>	Integer32	dbOptInternal	Internal
1.3.6.1.4.1.897.2.2.3.136.. <i>n</i>	String	dbOptJavaVmOptions*	java_vm_options
1.3.6.1.4.1.897.2.2.3.137.. <i>n</i>	Integer32	dbOptInternal	Internal
1.3.6.1.4.1.897.2.2.3.138.. <i>n</i>	Integer32	dbOptMaxClientStatementsC ached*	max_client_statements_c ached
1.3.6.1.4.1.897.2.2.3.139.. <i>n</i>	String	dbOptQueryMemTimeout*	query_mem_timeout
1.3.6.1.4.1.897.2.2.3.140.. <i>n</i>	String	dbOptAllowReadClientFile*	allow_read_client_file
1.3.6.1.4.1.897.2.2.3.141.. <i>n</i>	String	dbOptAllowWriteClientFile*	allow_write_client_file
1.3.6.1.4.1.897.2.2.3.142.. <i>n</i>	String	dbOptPriority*	priority
1.3.6.1.4.1.897.2.2.3.143.. <i>n</i>	String	dbOptMaxPriority*	max_priority

RDBMS MIB 参考

以下各节列出了可使用 SQL Anywhere SNMP Extension Agent 检索的值的 OID。缺省情况下，RDBMS MIB 位于 `C:\Program Files\SQL Anywhere 11\snmp\RDBMS-MIB.mib`。

rdbmsDbTable

此表列出了有关系统上安装的数据库的信息。

值 *db* 是 `sasnmplib.ini` 文件中的数据库编号。

OID	类型	名称	返回值
1.3.6.1.2.1.39.1.1.1.1. <i>db</i>	Integer	rdbmsDbIndex	<i>db</i>
1.3.6.1.2.1.39.1.1.1.2. <i>db</i>	OID	rdbmsDbPrivateMibOID	1.3.6.1.4.1.897.2
1.3.6.1.2.1.39.1.1.1.3. <i>db</i>	String	rdbmsDbVendorName	PROPERTY('CompanyName')
1.3.6.1.2.1.39.1.1.1.4. <i>db</i>	String	rdbmsDbName	DB_PROPERTY('Name')
1.3.6.1.2.1.39.1.1.1.5. <i>db</i>	String	rdbmsDbContact	PROPERTY('LicensedUser')

rdbmsDbInfoTable

此表提供有关系统上数据库的其它信息。

值 *db* 是 `sasnmplib.ini` 文件中的数据库编号。

OID	类型	名称	返回值
1.3.6.1.2.1.39.1.2.1.1. <i>db</i>	String	rdbmsDbInfoProductName	PROPERTY('ProductName')
1.3.6.1.2.1.39.1.2.1.2. <i>db</i>	String	rdbmsDbInfoVersion	PROPERTY('ProductVersion')

OID	类型	名称	返回值
1.3.6.1.2.1.39.1.2.1.3.db	Integer	rdbmsDbInfoSizeUnits	根据 dbInfoSizeAllocated 和 dbInfoSizeUsed 计算。 <ul style="list-style-type: none"> ● 1=字节 ● 2=KB ● 3=MB ● 4=GB ● 5=TB (每个单位均为上一个单位的 1024 倍。)
1.3.6.1.2.1.39.1.2.1.4.db	Integer	rdbmsDbInfoSizeAllocated	DB_PROPERTY('PageSize') * DB_PROPERTY('FileSize')
1.3.6.1.2.1.39.1.2.1.5.db	Integer	rdbmsDbInfoSizeUsed	DB_PROPERTY('PageSize') * (DB_PROPERTY('FileSize') - DB_PROPERTY('FreePages'))
1.3.6.1.2.1.39.1.2.1.6.db	String	rdbmsDbInfoLastBackup	NULL ¹

¹ SQL Anywhere SNMP Extension Agent 不支持此 OID。

rdbmsDbParamTable

此表列出了系统上数据库的配置参数。

此 *db* 是 *sasmp.ini* 文件中的数据库编号，而 *n* 是 *sa.2.3* 子树中的选项索引。

OID	类型	名称	返回值
1.3.6.1.2.1.39.1.3.1.1.db	String	rdbmsDbParamName	选项名称
1.3.6.1.2.1.39.1.3.1.2.db	Integer	rdbmsDbParamSubIndex	<i>n</i>
1.3.6.1.2.1.39.1.3.1.3.db	OID	rdbmsDbParamID	SQL Anywhere MIB 中与此选项对应的 OID
1.3.6.1.2.1.39.1.3.1.4.db	String	rdbmsDbParamCurrValue	选项值
1.3.6.1.2.1.39.1.3.1.5.db	String	rdbmsDbParamComment	NULL ¹

¹ SQL Anywhere SNMP Extension Agent 不支持此 OID。

rdbmsDbLimitedResourceTable

此表列出了关于每个 dbspace 的空闲空间信息。在此表中，*n* 代表各个 dbspace，如下所示：

- 1-13 代表常规 dbspace（在数据库中的编号为 0-12）
- 14 是事务日志文件
- 15 是事务日志镜像文件
- 16 是临时文件

值 *db* 是 *sasnmplib.ini* 文件中的数据库编号。

OID	类型	名称	返回值
1.3.6.1.2.1.39.1.4.1.1. <i>n.d</i> <i>b</i>	String	rdbmsDbLimitedResourceName	dbspace 的名称或者事务日志、事务日志镜像或临时文件的名称。
1.3.6.1.2.1.39.1.4.1.2. <i>n.d</i> <i>b</i>	OID	rdbmsDbLimitedResourceID	1.3.6.1.4.1.897.2
1.3.6.1.2.1.39.1.4.1.3. <i>n.d</i> <i>b</i>	Integer	rdbmsDbLimitedResourceLimit	磁盘上可用的空闲空间 + 当前文件大小
1.3.6.1.2.1.39.1.4.1.4. <i>n.d</i> <i>b</i>	Integer	rdbmsDbLimitedResourceCurrent	当前文件大小
1.3.6.1.2.1.39.1.4.1.5. <i>n.d</i> <i>b</i>	Integer	rdbmsDbLimitedResourceHighwater	当前大小
1.3.6.1.2.1.39.1.4.1.6. <i>n.d</i> <i>b</i>	Integer	rdbmsDbLimitedResourceFailure	0 ¹
1.3.6.1.2.1.39.1.4.1.7. <i>n.d</i> <i>b</i>	String	rdbmsDbLimitedResourceDescription	字节、KB、MB、GB 或 TB 之一。

¹ SQL Anywhere SNMP Extension Agent 不支持此 OID。

rdbmsSrvTable

此表列出了系统上运行或安装的数据库服务器。

值 *db* 是 *sasnmplib.ini* 文件中的数据库编号。

OID	类型	名称	返回值
1.3.6.1.2.1.39.1.5.1.1. <i>db</i>	OID	rdbmsSrvPrivateMibOID	1.3.6.1.4.1.897.2

OID	类型	名称	返回值
1.3.6.1.2.1.39.1.5.1.2.db	String	rdbmsSrvVendorName	PROPERTY('CompanyName')
1.3.6.1.2.1.39.1.5.1.3.db	String	rdbmsSrvProductName	PROPERTY('ProductName')
1.3.6.1.2.1.39.1.5.1.4.db	String	rdbmsSrvContact	PROPERTY('LicensedCompany')

rdbmsSrvInfoTable

此表列出了有关系统中数据库服务器的其它信息。

值 *db* 是 *sasnmplib.ini* 文件中的数据库编号。

OID	类型	名称	返回值
1.3.6.1.2.1.39.1.6.1.1.db	Integer	rdbmsSrvInfoStartupTime	PROPERTY('StartTime')
1.3.6.1.2.1.39.1.6.1.2.db	Integer	rdbmsSrvInfoFinishedTransactions	0 ¹
1.3.6.1.2.1.39.1.6.1.3.db	Integer	rdbmsSrvInfoDiskReads	PROPERTY('DiskReadEng')
1.3.6.1.2.1.39.1.6.1.4.db	Integer	rdbmsSrvInfoLogicalReads	0 ¹
1.3.6.1.2.1.39.1.6.1.5.db	Integer	rdbmsSrvInfoDiskWrites	PROPERTY('DiskWriteEng')
1.3.6.1.2.1.39.1.6.1.6.db	Integer	rdbmsSrvInfoLogicalWrites	0 ¹
1.3.6.1.2.1.39.1.6.1.7.db	Integer	rdbmsSrvInfoPageReads	0 ¹
1.3.6.1.2.1.39.1.6.1.8.db	Integer	rdbmsSrvInfoPageDiskOutOfWrites	0 ¹
1.3.6.1.2.1.39.1.6.1.9.db	Integer	rdbmsSrvInfoSpaces	0 ¹
1.3.6.1.2.1.39.1.6.1.10.db	Integer	rdbmsSrvInfoHandledRequests	PROPERTY('Req')
1.3.6.1.2.1.39.1.6.1.11.db	Integer	rdbmsSrvInfoRequestRecvs	PROPERTY('PacketsReceivedUncomp')
1.3.6.1.2.1.39.1.6.1.12.db	Integer	rdbmsSrvInfoRequestSends	PROPERTY('PacketsSentUncomp')
1.3.6.1.2.1.39.1.6.1.13.db	Integer	rdbmsSrvInfoHighwaterInboundAssociations	0 ¹

OID	类型	名称	返回值
1.3.6.1.2.1.39.1.6.1.14.db	Integer	rdbmsSrvInfoMaxInboundAssociations	0 ¹

¹ SQL Anywhere SNMP Extension Agent 不支持此 OID。

rdbmsSrvParamTable

此表列出了可由 SQL Anywhere SNMP Extension Agent 通过 SQL Anywhere MIB 设置的服务器选项。*n* 是索引，如下所示：

<i>n</i>	服务器选项
1	ConnsDisabled
2	LivenessTimeout (缺省值)
3	QuittingTime
4	RememberLastStatement
5	RequestLogFile
6	RequestLogging

值 *db* 是 *sasnmplib.ini* 文件中的数据库编号。

OID	类型	名称	返回值
1.3.6.1.2.1.39.1.7.1.1.n.db	String	rdbmsDbSrvParamName	选项 <i>n</i> 的名称
1.3.6.1.2.1.39.1.7.1.2.n.db	Integer	rdbmsDbSrvParamSubIndex	<i>n</i>
1.3.6.1.2.1.39.1.7.1.3.n.db	OID	rdbmsDbSrvParamID	1.3.6.1.4.1.897.2
1.3.6.1.2.1.39.1.7.1.4.n.db	String	rdbmsDbSrvParamCurrentValue	选项 <i>n</i> 的当前值
1.3.6.1.2.1.39.1.7.1.5.n.db	String	rdbmsDbSrvParamComment	选项 <i>n</i> 的完整名称

rdbmsSrvLimitedResourceTable

此表包含有关服务器配置参数的信息。

值 *db* 是 *sasnmplib.ini* 文件中的数据库编号，*n* 是资源索引，如下所示：

<i>n</i>	名称	资源	资源限制
1	连接	PROPERTY('UniqueClientAddresses')	PROPERTY('LicenseCount')
2	处理器	PROPERTY('NumLogicalProcessorsUsed')	PROPERTY('NumLogicalProcessorsUsed')

OID	类型	名称	返回值
1.3.6.1.2.1.39.1.8.1.1. <i>db</i>	String	rdbmsSrvLimitedResourceName	资源 <i>n</i> 的名称
1.3.6.1.2.1.39.1.8.1.2. <i>db</i>	OID	rdbmsSrvLimitedResourceID	SQL Anywhere MIB 中与此选项对应的 OID
1.3.6.1.2.1.39.1.8.1.3. <i>db</i>	Integer	rdbmsSrvLimitedResourceLimit	资源 <i>n</i> 的上限
1.3.6.1.2.1.39.1.8.1.4. <i>db</i>	Integer	rdbmsSrvLimitedResourceCurrent	资源 <i>n</i> 的当前值
1.3.6.1.2.1.39.1.8.1.5. <i>db</i>	Integer	rdbmsSrvLimitedResourceHighwater	资源 <i>n</i> 的当前值
1.3.6.1.2.1.39.1.8.1.6. <i>db</i>	Integer	rdbmsSrvLimitedResourceFailures	0 ¹
1.3.6.1.2.1.39.1.8.1.7. <i>db</i>	String	rdbmsSrvLimitedResourceDescription	资源 <i>n</i> 的名称

¹ SQL Anywhere SNMP Extension Agent 不支持此 OID。

安全

本节介绍 SQL Anywhere 的安全功能。

保护数据的安全	1003
传送层安全	1031

保护数据的安全

目录

安全功能简介	1004
安全提示	1005
控制数据库访问	1006
审计数据库活动	1012
以安全的方式运行数据库服务器	1018
加密和解密数据库	1019
保护 Windows Mobile 数据库的安全	1029

安全功能简介

由于数据库可能包含专有、机密或专用信息，因此确保数据库及其中数据采用安全保护设计非常重要。

SQL Anywhere 提供了几个有助于为数据建立安全环境的功能：

- **用户标识和验证** 这些功能控制哪些人拥有数据库访问权限。请参见“[创建新用户](#)”一节第 430 页。
- **自由访问控制功能** 这些功能控制用户在连接到数据库的情况下可以执行的操作。请参见“[数据库权限和特权概述](#)”一节第 422 页。
- **审计** 此功能可帮助您建立一份数据库操作的记录。请参见“[审计数据库活动](#)”一节第 1012 页。
- **数据库服务器选项** 这些功能用于控制可以执行管理操作（例如，装载数据库）的人。这些选项是在启动数据库服务器时设置的。请参见“[从命令行控制权限](#)”一节第 45 页。
- **视图和存储过程** 这些功能用于指定用户可以访问的数据及用户可以执行的操作。请参见“[使用视图和过程来获得更高的安全性](#)”一节第 452 页。
- **数据库和表加密** 可以选择使用简单加密或高度加密来保护数据库的安全。简单加密等效于模糊处理。而通过高度加密，在没有加密密钥的情况下，数据库将完全无法访问。请参见“[-ek 数据库选项](#)”一节第 238 页和“[DatabaseKey 连接参数 \[DBKEY\]](#)”一节第 259 页。
表加密功能用于加密单个表，而不是整个数据库。请参见“[表加密](#)”一节第 1026 页。
- **传送层安全** 可以使用传送层安全来验证客户端应用程序与数据库服务器之间的通信。传送层安全使用椭圆曲线或 RSA 加密技术。请参见“[传送层安全](#)”第 1031 页。

注意

如果担心运行数据库服务器的计算机上的其它进程能够访问客户端/服务器通信的内容，则建议使用加密。

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](#)》。

- **受保护的功能** 可以为数据库服务器上运行的所有数据库禁用功能。
- **SELinux 支持** SELinux 策略控制应用程序对系统资源的访问。SQL Anywhere 包括在 Red Hat Enterprise Linux 5 上保护其安全的策略。

有关编译和安装 SQL Anywhere SELinux 策略的信息，请参见 *install-dir/selinux/readme*。

数据安全由数据库管理员负责。在本章中，除非另有说明，否则必须拥有 DBA 权限才能执行所述的任务。

用户 ID 和权限是与安全有关的主题。请参见“[管理用户 ID、特权和权限](#)”第 415 页。

安全提示

作为数据库管理员，可以采取许多措施来提高数据的安全性。例如，可以采取以下措施：

- **谨慎选择口令** 不要部署使用缺省用户 ID 和口令的数据库。请参见“[提高口令安全性](#)”一节第 1006 页。
- **限制 DBA 权限** 因为 DBA 权限非常强大，所以应仅限确实需要该权限的用户使用。具有 DBA 权限的用户可以查看数据库中的任何内容，可以在数据库中执行任何操作。

可以考虑为具有 DBA 权限的用户提供两个用户 ID：一个具有 DBA 权限，一个不具有该权限，这样他们就可以只在有必要时才以 DBA 权限用户的身份进行连接。
- **使用受保护的数据库功能** 数据库服务器 -sf 选项用于为数据库服务器上运行的所有数据库启用和禁用功能。可以禁用的功能包括使用外部存储过程、Java、远程数据访问及更改请求日志设置的功能。请参见“[-sf 服务器选项](#)”一节第 207 页和“[指定受保护的功能](#)”一节第 1010 页。
- **删除外部系统函数** 以下外部函数可能会带来安全风险：xp_cmdshell、xp_startmail、xp_startsmtp、xp_sendmail、xp_stopmail 和 xp_stopsmtp。

xp_cmdshell 过程允许用户执行操作系统命令或程序。

电子邮件命令允许用户指示服务器发送由用户撰写的电子邮件。恶意用户可能会使用电子邮件或命令 shell 过程，以操作系统所授予权限之外的其它权限执行操作系统任务。在注重安全的环境中，应该删除这些函数。

有关删除过程的信息，请参见“[DROP PROCEDURE 语句](#)”一节《[SQL Anywhere 服务器 - SQL 参考](#)》。
- **保护数据库文件** 应该防止他人对数据库文件、日志文件和 dbspace 文件进行未经授权的访问。不要将它们存储在共享目录或卷中。
- **保护数据库软件** 也应以类似方式保护 SQL Anywhere 软件。只授予用户访问其所需的应用程序、DLL 及其它资源的权限。
- **将数据库服务器作为服务或守护程序运行** 为防止未经授权的用户关闭数据库或获得对数据库或日志文件的访问权限，请将数据库服务器作为 Windows 服务运行。在 Unix 上，将服务器作为守护程序运行可起到相似作用。请参见“[在当前会话外部运行服务器](#)”一节第 57 页。
- **将 SATMP 设置为某个唯一目录** 要在 Unix 平台上保护数据库服务器的安全，请将 SATMP 设置为某个唯一目录，并对该目录实施读、写及执行操作保护，防止所有其他用户对其执行这些操作。这样做会强制所有其它连接使用 TCP/IP，其安全性要高于共享内存连接。

在客户端与服务器之间发送任何实际数据前，二者之间使用的共享内存缓冲区将从目录树中删除。这意味着其它进程无法看到任何通信数据，这是由于共享内存缓冲区/文件被隐藏，因而进程无法获取其句柄。
- **对数据库进行高度加密** 如果对数据库进行高度加密，则在不使用密钥的情况下数据库将完全无法访问。使用任何其它手段都无法打开数据库或查看数据库或事务日志文件。

有关详细信息，请参见“[-ep 服务器选项](#)”一节第 173 页和“[-ek 数据库选项](#)”一节第 238 页。

控制数据库访问

通过指派用户 ID 和口令，数据库管理员可以控制哪些人拥有数据库访问权限。通过给每个用户 ID 授予权限，数据库管理员可以控制每个用户连接到数据库时可以执行的任务。

权限模式基于用户 ID

用户登录数据库时，可以访问符合以下任一条件的所有数据库对象：

- 用户创建的对象
- 用户获得了显式访问权限的对象
- 用户所属组获得了显式访问权限的对象

用户无法访问任何不符合上述条件的数据库对象。简言之，用户只能访问其拥有的对象或显式获得了访问权限的对象。

有关详细信息，请参见：

- “管理用户 ID、特权和权限” 第 415 页
- “CONNECT 语句 [ESQL] [Interactive SQL]” 一节 《SQL Anywhere 服务器 - SQL 参考》
- “GRANT 语句” 一节 《SQL Anywhere 服务器 - SQL 参考》
- “REVOKE 语句” 一节 《SQL Anywhere 服务器 - SQL 参考》

使用集成登录

集成登录允许用户使用单一登录名和口令登录 Windows 操作系统和数据库。外部登录名与数据库用户 ID 相关联。在尝试进行集成登录时，用户通过提供登录名和口令来登录操作系统。操作系统随即将用户身份告知服务器，服务器将允许用户以关联的数据库用户 ID 身份登录数据库，而不会再要求用户输入登录名或口令。

使用集成登录时，如果保持用户配置文件 Guest 的启用状态不变且不设置任何口令，用户将可对接受集成登录的服务器所代管的数据库进行不受限制的访问。确切地讲，任何用户都可以使用任意登录 ID 和任意口令登录该服务器，因为缺省情况下他们会登录到 Guest 用户配置文件。

有关详细信息，请参见：

- “安全问题：不受限制的数据库访问” 一节第 105 页
- “使用集成登录” 一节第 99 页
- “login_mode 选项 [数据库]” 一节第 512 页

提高口令安全性

口令在任何数据库安全系统中都是一个重要部分。要做到安全，口令必须不容易被猜到，并且必须难以在用户的硬盘或其它位置得到。SQL Anywhere 口令始终区分大小写。可以使用 `verify_password_function` 选项指定用于口令验证的函数。请参见 “`verify_password_function` 选项 [数据库]” 一节第 560 页。

实现登录策略

使用登录策略控制用户口令更改的频率以及指定锁定帐户之前所允许的登录尝试次数。请参见“[管理登录策略概述](#)”一节第 416 页或“[CREATE LOGIN POLICY 语句](#)”一节《[SQL Anywhere 服务器 - SQL 参考](#)》。

更改缺省用户 ID 和口令

新创建数据库的缺省用户 ID 为 **DBA**，口令为 **sql**。应在部署数据库之前更改此口令。

实现最低口令长度

缺省情况下，对口令的长度没有任何限制。要实现更高的安全性，可以对所有新口令规定最小长度要求，不允许使用短（因而容易被猜到的）口令。可通过将 `min_password_length` 数据库选项设置为一个大于零的值来实现此目的。下列语句强制口令至少为 8 字节长。

```
SET OPTION PUBLIC.min_password_length = 8;
```

请参见“[min_password_length 选项 \[数据库\]](#)”一节第 522 页。

实现口令到期功能

缺省情况下，数据库口令永远不到期。可以使用登录策略来实现口令到期功能。请参见“[管理登录策略概述](#)”一节第 416 页。

不要在 ODBC 数据源中包含口令

口令是用于访问数据库的密钥。在注重安全的环境中，口令不应被未经授权的人轻易地获取。

创建 ODBC 数据源或 Sybase Central 连接配置文件时，可以选择是否包括口令。请避免包括口令，以确保口令不会被未经授权的用户看到。

请参见“[创建 ODBC 数据源](#)”一节第 90 页。

对包含口令的配置文件进行加密

创建配置文件时，可以选择是否包括口令信息。为保护口令，可以考虑使用文件隐藏 (`dbfhide`) 实用程序通过简单加密来隐藏配置文件的内容。请参见“[文件隐藏实用程序 \(dbfhide\)](#)”一节第 724 页。

使用口令验证

可以使用 `verify_password_function` 选项来指定实现口令规则的函数。请参见“[verify_password_function 选项 \[数据库\]](#)”一节第 560 页。

以下示例定义表和函数并设置一些登录策略选项。它们共同实现了高级口令规则，其中包括要求在口令中使用某些类型的字符、不允许口令重用和为口令设置有效期。当创建用户 ID 或更改口令时，数据库服务器会通过 `verify_password_function` 选项调用函数。应用程序可以调用 `post_login_procedure` 选项指定的过程，以报告口令应在到期前进行更改。

也可以从以下位置获得此示例的代码：`samples-dir\SQLAnywhere\SQL\verify_password.sql`。（有关 `samples-dir` 的信息，请参见“[示例目录](#)”一节第 368 页。）

```

-- This example defines a function that implements advanced password rules
-- including requiring certain types of characters in the password and
-- disallowing password reuse. The f_verify_pwd function is called by the
-- server using the verify_password_function option when a user ID is
-- created or a password is changed.
--
-- The "root" login profile is configured to expire passwords every 180 days
-- and lock non-DBA accounts after 5 consecutive failed login attempts.
--
-- The application may call the procedure specified by the
-- post_login_procedure option to report that the password should be changed
-- before it expires.

-- only DBA should have permissions on this table
CREATE TABLE DBA.t_pwd_history(
    pk          INT          DEFAULT AUTOINCREMENT PRIMARY KEY,
    user_name   CHAR(128),   -- the user whose password is set
    pwd_hash    CHAR(32) );  -- hash of password value to detect
                            -- duplicate passwords

-- called whenever a non-NULL password is set
-- to verify the password conforms to password rules
CREATE FUNCTION DBA.f_verify_pwd( uid      VARCHAR(128),
                                new_pwd VARCHAR(255) )
RETURNS VARCHAR(255)
BEGIN
    -- a table with one row per character in new_pwd
    DECLARE local temporary table pwd_chars(
        pos INT PRIMARY KEY,    -- index of c in new_pwd
        c   CHAR( 1 CHAR ) );  -- character
    -- new_pwd with non-alpha characters removed
    DECLARE pwd_alpha_only   CHAR(255);
    DECLARE num_lower_chars  INT;

    -- enforce minimum length (can also be done with
    -- min_password_length option)
    IF length( new_pwd ) < 6 THEN
        RETURN 'password must be at least 6 characters long';
    END IF;

    -- break new_pwd into one row per character
    INSERT INTO pwd_chars SELECT row_num, substr( new_pwd, row_num, 1 )
        FROM dbo.RowGenerator
        WHERE row_num <= length( new_pwd );

    -- copy of new_pwd containing alpha-only characters
    SELECT list( c, ' ORDER BY pos ) INTO pwd_alpha_only
        FROM pwd_chars WHERE c BETWEEN 'a' AND 'z' OR c BETWEEN 'A' AND 'Z';

    -- number of lower case characters IN new_pwd
    SELECT count(*) INTO num_lower_chars
        FROM pwd_chars WHERE CAST( c AS BINARY ) BETWEEN 'a' AND 'z';

    -- enforce rules based on characters contained in new_pwd
    IF ( SELECT count(*) FROM pwd_chars WHERE c BETWEEN '0' AND '9' )
        < 1 THEN
        RETURN 'password must contain at least one numeric digit';
    ELSEIF length( pwd_alpha_only ) < 2 THEN
        RETURN 'password must contain at least two letters';
    ELSEIF num_lower_chars = 0
        OR length( pwd_alpha_only ) - num_lower_chars = 0 THEN
        RETURN 'password must contain both upper- and lowercase characters';
    END IF;

```

```

-- not the same as any user name
-- (this could be modified to check against a disallowed words table)
IF EXISTS( SELECT * FROM SYS.SYSUSER
           WHERE lower( user_name ) IN ( lower( pwd_alpha_only ),
                                         lower( new_pwd ) ) ) THEN
    RETURN 'password or only alphabetic characters in password ' ||
           'must not match any user name';
END IF;

-- not the same as any previous password for this user
IF EXISTS( SELECT * FROM t_pwd_history
           WHERE user_name = uid
             AND pwd_hash = hash( uid || new_pwd, 'md5' ) ) THEN
    RETURN 'previous passwords cannot be reused';
END IF;

-- save the new password
INSERT INTO t_pwd_history( user_name, pwd_hash )
VALUES( uid, hash( uid || new_pwd, 'md5' ) );

RETURN( NULL );
END;

ALTER FUNCTION DBA.f_verify_pwd SET HIDDEN;
GRANT EXECUTE ON DBA.f_verify_pwd TO PUBLIC;
SET OPTION PUBLIC.verify_password_function = 'DBA.f_verify_pwd';

-- All passwords expire in 180 days. Expired passwords can be changed
-- by the user using the NewPassword connection parameter.
ALTER LOGIN POLICY root password_life_time = 180;

-- If an application calls the procedure specified by the
-- post_login_procedure option, then the procedure can be used to warn
-- the user that their password is about to expire. In particular,
-- Interactive SQL and Sybase Central call the
-- post_login_procedure system procedure.
ALTER LOGIN POLICY root password_grace_time = 30;

-- Five consecutive failed login attempts will result in a non-DBA
-- user ID being locked.
ALTER LOGIN POLICY root max_failed_login_attempts = 5;

```

控制用户可以执行的任务

可以通过授予许可来控制用户能够对数据库对象执行的任务（如创建、修改、执行和更新等等）。可以通过授权来控制用户能够执行的管理任务（如备份和分析等等）。

使用 GRANT 语句授予许可和权限。对于许可，还可以将针对某个对象的许可授予特权委派给其他用户。

REVOKE 语句与 GRANT 语句的作用正好相反—GRANT 语句显式授予的任何许可都能使用 REVOKE 语句进行撤消。撤消用户的 CONNECT 权限会将该用户及其拥有的所有对象从数据库中删除。

另请参见

- “管理用户 ID、特权和权限” 第 415 页
- “GRANT 语句” 一节 《SQL Anywhere 服务器 - SQL 参考》
- “REVOKE 语句” 一节 《SQL Anywhere 服务器 - SQL 参考》

设计安全的数据库对象

视图和存储过程提供了一些替代方法，这些方法用于对用户可以访问的数据和可以执行的任务进行调整。

请参见：

- “过程和触发器的优点” 一节 《SQL Anywhere 服务器 - SQL 的用法》
- “使用视图和过程来获得更高的安全性” 一节 第 452 页

指定受保护的功能

要控制用户可以使用的数据库功能，可以在启动数据库服务器时包括受保护的功能选项 (-sf)。该受保护的功能选项控制以下类别功能的可用性：

- 服务器端备份
- 外部存储过程
- 远程数据访问
- Web 服务

有关功能的完整列表，请参见 “-sf 服务器选项” 一节 第 207 页。

还可以在启动数据库服务器时包括 -sk 选项。此选项指定可用于为特定连接重新启用受保护功能的密钥。为连接重新启用受保护功能的方法是，将 secure_feature_key 临时选项的值设置为启动数据库服务器时 -sk 指定的值。

要修改连接的受保护功能或功能集，请用 -sk 指定一个密钥并将 secure_feature_key 临时选项设置为该密钥值，以使用 sa_server_option 系统过程。为启用或禁用功能而进行的任何更改都会立即生效。

◆ 保护数据库功能的安全

1. 使用 -sf 和 -sk（可选）选项启动数据库服务器。

例如，以下命令会启动数据库服务器并禁用远程数据访问。不过，它包含一个密钥，可用来为连接重新启用已禁用的功能。

```
dbsrv11 -n secure_server -sf remote_data_access -sk ls64uwq15 c:\mydata.db
```

2. 连接到数据库服务器。

例如：

```
dbisql -c "UID=DBA;PWD=sql;ENG=secure_server;DBN=demo"
```

3. 将 secure_feature_key 临时选项的值设置为启动数据库服务器时 -sk 指定的值。

例如：

```
SET TEMPORARY OPTION secure_feature_key = 'ls64uwq15';
```

4. 使用 `sa_server_option` 系统过程更改数据库服务器的受保护功能。

例如：

```
CALL sa_server_option( 'SecureFeatures', '-remote_data_access' );
```

另请参见

- “-sf 服务器选项” 一节第 207 页
- “-sk 服务器选项” 一节第 210 页
- “secure_feature_key [数据库]” 一节第 544 页
- “sa_server_option 系统过程” 一节 《SQL Anywhere 服务器 - SQL 参考》

审计数据库活动

每个数据库都有一个与之关联的事务日志文件。事务日志用于数据库恢复。它是有关针对数据库执行的事务的记录。请参见“[事务日志](#)”一节第 14 页。

事务日志存储所有已执行的数据定义语句和执行这些语句的用户 ID。它还存储所有的更新、删除、插入以及执行那些语句的用户。然而，对某些审计目的而言，这是不够的。缺省情况下，事务日志并不包含事件的发生时间，而只是包含事件发生的顺序。此外，它既不包含失败的事件，也不包含 SELECT 语句。

审计 是跟踪对数据库所执行活动的一种方法。使用审计时，会在事务日志中保存附加数据，其中包括：

- 包括终端 ID 在内的所有登录尝试（成功的和失败的）。
- 所有事件的准确时间戳（精确到毫秒）。
- 所有的权限检查（成功的和失败的），包括检查的权限所涉及的对象（如果适用）。
- 所有要求具有 DBA 权限的操作。

为数据库启用了审计时，无法停止使用事务日志。如果要禁用事务日志，必须先禁用审计。

控制审计

数据库管理员可以启用审计，以向事务日志添加与安全有关的信息。这可以通过使用 Sybase Central 或 Interactive SQL 来完成。

缺省情况下，审计处于禁用状态。要启用和禁用审计，必须具有 DBA 权限。

◆ 控制审计 (Sybase Central)

1. 以具有 DBA 权限的用户身份连接到数据库。
2. 右击数据库并选择 **[属性]**。
3. 单击 **[审计]** 选项卡并选择下列选项之一：
 - **不收集此数据库的审计信息** 不收集任何审计信息。此选项通过将审计数据库选项设置为 Off 来禁用审计。请参见“[auditing 选项 \[数据库\]](#)”一节第 484 页。
 - **收集此数据库的所有审计信息** 收集数据库的所有类型审计信息。此选项通过将审计数据库选项设置为 On 来启用审计。请参见“[auditing 选项 \[数据库\]](#)”一节第 484 页。
事务日志在选择此选项时可能显著增大。
 - **收集此数据库的以下类型的审计信息** 允许指定需要收集的审计信息。例如，可以选择仅收集 DDL 更改。请参见“[sa_enable_auditing_type 系统过程](#)”一节《[SQL Anywhere 服务器 - SQL 参考](#)》。选择此选项将更改 auditing_options 数据库选项的设置。请参见“[auditing_options 选项 \[数据库\]](#)”一节第 485 页。
4. 单击 **[确定]**。

◆ 控制审计 (Interactive SQL)

1. 以具有 DBA 权限的用户身份连接到数据库。
2. 执行以下语句启用审计：

```
SET OPTION PUBLIC.auditing = 'On';
```

要指定想要启用的审计信息类型，请使用以下系统过程：

```
CALL sa_enable_auditing_type( 'all' );
```

可以通过使用想要启用的审计类型替换 **all** 来控制收集的审计信息的类型。请参见“[sa_enable_auditing_type 系统过程](#)”一节《[SQL Anywhere 服务器 - SQL 参考](#)》。

3. 执行以下语句关闭审计：

```
SET OPTION PUBLIC.auditing = 'Off';
```

要指定想要禁用的审计信息类型，请使用以下系统过程：

```
CALL sa_disable_auditing_type( 'all' );
```

可以通过使用想要禁用的审计类型替换 **all** 来停止收集指定类型的审计信息。请参见“[sa_disable_auditing_type 系统过程](#)”一节《[SQL Anywhere 服务器 - SQL 参考](#)》。

审计个别连接

为数据库启用了审计后，即可在数据库登录过程中设置 `conn_auditing` 临时数据库选项来启用特定于连接的审计。可以根据客户端计算机 IP 地址或连接类型之类的信息启用审计。

如果在登录过程中不设置 `conn_auditing` 选项，缺省情况下该选项将为启用状态。

以下示例显示登录过程的一个片段，该片段为除 DBA 用户进行的连接之外的所有数据库连接启用审计：

```
DECLARE usr VARCHAR(128)
SELECT CONNECTION_PROPERTY( 'Userid' ) INTO usr;
IF usr != 'DBA' THEN
    SET TEMPORARY OPTION conn_auditing='On'
ELSE
    SET TEMPORARY OPTION conn_auditing='Off'
END IF;
```

有关详细信息，请参见“[login_procedure 选项 \[数据库\]](#)”一节第 513 页和“[conn_auditing 选项 \[数据库\]](#)”一节第 491 页。

另请参见

- “[auditing 选项 \[数据库\]](#)”一节第 484 页

检索审计信息

可以使用 Sybase Central 或翻译日志文件实用程序 (dbtran) 来从事务日志检索审计信息。尝试检索审计信息前，要确保以具有 DBA、REMOTE 或 BACKUP 权限的用户身份连接到数据库。

◆ 检索审计信息 (Sybase Central)

1. 选择数据库。
2. 单击 [审计] 选项卡。
3. 单击 [检索审计消息]。

将出现一个窗口，其中显示 dbtran 消息。忽略有关按时间顺序输出的警告。

4. 单击 [关闭]。

审计信息出现在右窗格中的 [审计] 选项卡上。

5. 使用过滤器选项控制想要显示的审计信息。

可以选择显示所有审计信息，或者仅显示错误或包含指定文本的审计消息。

6. 选择审计条目表中的某个条目以显示有关此条目的详细信息。

7. 要检索最新审计信息，请按 F5 键，然后重复上述过程。

有关详细信息，请参见“[审计示例](#)”一节第 1015 页。

使用 dbtran 实用程序检索审计信息

可以从 Sybase Central 或命令提示符访问 dbtran 实用程序。dbtran 实用程序使用指定的事务日志来生成一个 SQL 脚本，该脚本包含所有事务以及关于执行每条命令的用户的一些信息。通过使用 -g 选项，dbtran 可以包括更多包含审计信息的注释。-g 选项等效于指定以下选项：

- -d 按时间顺序显示输出。
- -t 在输出中包含触发器生成的操作。
- -a 在输出中包含回退的事务。

有关这些选项的详细信息，请参见“[翻译日志文件实用程序 \(dbtran\)](#)”一节第 753 页。

可以对正在运行的数据库服务器或数据库日志文件运行 dbtran 实用程序。

◆ 从正在运行的数据库服务器检索审计信息

- 在数据库服务器运行时，运行以下命令：

```
dbtran -g -c connection-string -n SQL-file
```

例如：

```
dbtran -g -c "UID=DBA;PWD=sql" -n demo.sql
```

事务日志的可读版本被保存到用户的当前目录。在此示例中，审计信息被保存到 *demo.sql* 文件，此文件包含有关示例数据库的信息。

有关连接字符串的详细信息，请参见“[连接参数](#)”一节第 248 页。

◆ 从事务日志文件中检索审计信息

1. 关闭数据库服务器以确保可以使用事务日志文件。
2. 运行以下命令：

```
dbtran -g transaction-log SQL-file
```

例如：

```
dbtran -g demo.log demo.sql
```

在此示例中，来自事务日志文件 *demo.log* 的审计信息会放入文件 *demo.sql*。

有关详细信息，请参见“[翻译日志文件实用程序 \(dbtran\)](#)”一节第 753 页。

添加审计注释

可以使用 `sa_audit_string` 系统存储过程向审计跟踪信息中添加注释。它使用单一参数，该参数是一个最多为 200 字节的字符串。要调用此过程，必须具有 DBA 权限。

例如：

```
CALL sa_audit_string( 'Started audit testing here.' );
```

此注释以审计语句形式存储在事务日志中。

审计示例

此示例使用 Sybase Central 或 Interactive SQL 说明审计功能如何记录那些访问未授权信息的尝试。

◆ 审计示例 (Sybase Central)

1. 启动 Sybase Central 并使用 SQL Anywhere 11 Demo 数据源连接到示例数据库。
这会使您以 DBA 用户身份进行连接。
2. 启用审计：
 - a. 右击数据库并选择 [属性]。
 - b. 单击 [审计] 选项卡。
 - c. 单击 [收集此数据库的所有审计信息]
 - d. 单击 [应用]。
 - e. 单击 [确定]。
3. 将名为 Test1 的用户添加到示例数据库，口令为 welcome：
 - a. 右击 [用户和组]，然后选择 [新建] » [用户]。
 - b. 当出现提示时，命名用户 **Test1**，然后键入他们的口令 **welcome**。

- c. 为用户提供 [分析权限]。
 - d. 单击 [完成]。
 - e. 断开与示例数据库的连接。
4. 使用 Sybase Central，以 Test1 身份连接到示例数据库，并尝试访问 Employees 表中的机密信息：
 - a. 选择 [表]，然后选择 Employees 表。
 - b. 单击 [数据] 选项卡。

出现一条错误消息：[权限被拒绝：您没有从 "Employees" 中进行选择的权限]。
 - c. 单击 [确定]。
 - d. 断开与示例数据库的连接。
 5. 查看此活动的审计信息：
 - a. 使用 Sybase Central，以具有 DBA 权限的用户身份连接到示例数据库。
 - b. 选择该数据库，然后单击右窗格中的 [审计] 选项卡。
 - c. 单击 [检索审计消息]。
 - d. 单击 [关闭]。

将出现审计信息。
 - e. 使用过滤器选项在审计信息表中查找错误。可以通过选择 [仅错误] 选项找到 BadUser 的错误。使用日期和时间信息精确定位错误。例如，如果 BadUser 在 2007 年 11 月 6 日的 10:07:14 尝试访问 Employees 表，则对应审计条目类似以下条目：

```
2007-11-06 10:07:14 | Permission
```
 6. 将示例数据库恢复至其初始状态：
 - a. 右击该数据库，然后选择 [属性]。
 - b. 在 [审计] 选项卡上，选择 [不收集此数据库的审计信息]。
 - c. 单击 [确定]。
 - d. 选择 [用户和组]。

右击 Test1，然后选择 [删除]。

◆ 审计示例 (Interactive SQL)

1. 启动 Interactive SQL 并使用 SQL Anywhere 11 Demo 数据源连接到示例数据库。

这会使您以 DBA 用户身份进行连接。
2. 使用 SET OPTION 语句启用审计，如下所示：
3. 使用 CREATE USER 语句将用户 Test1 添加到示例数据库，如下所示：

```
SET OPTION PUBLIC.auditing = 'On';
```

```
CREATE USER Test1  
IDENTIFIED BY welcome;
```

4. 打开一个新的 Interactive SQL 窗口，以 BadUser 身份连接到示例数据库，并尝试使用以下 SELECT 语句访问 Employees 表中的机密信息：

```
SELECT Surname, Salary
FROM GROUPO.Employees;
```

您会收到一条错误消息：[权限被拒绝：您没有从 "Employees" 中进行选择的权限]。

5. 运行以下命令查看此活动的审计信息：

```
dbtran -g -c "DSN=SQL Anywhere 11 Demo" -n demo.sql
```

6. 将示例数据库恢复至其初始状态：

- 使用 DROP USER 语句从数据库删除 Test1 用户：

```
DROP USER Test1;
```

- 使用以下 SET OPTION 语句关闭审计：

```
SET OPTION PUBLIC.auditing = 'Off';
```

数据库服务器外的审计操作

一些数据库实用程序直接作用于数据库文件。在一个安全的环境中，应该只有受信任的用户才有权访问数据库文件。

在 Windows 或 Unix 中为了对操作进行审计，所有使用 dbtran 或 dblog 的操作都会生成一个文本文件，该文件与数据库文件位于同一目录，扩展名为 .alg。例如，如果数据库文件为 demo.db，则该文件的名称将是 demo.alg。包含工具名、Windows 或 Unix 用户名以及日期/时间的记录将被追加到该文件中。仅当审计选项设置为 On 时，记录才会被追加到 .alg 文件中。

另请参见

- [“auditing 选项 \[数据库\]”一节第 484 页](#)
- [“翻译日志文件实用程序 \(dbtran\)”一节第 753 页](#)
- [“事务日志实用程序 \(dblog\)”一节第 792 页](#)

以安全的方式运行数据库服务器

有几种安全功能您可以在启动数据库服务器时，或者在服务器运行过程中设置，其中包括：

- **启动和停止数据库** 使用个人数据库服务器时，缺省情况下任何用户都可以在运行着的服务器上启动其它数据库。缺省情况下，网络数据库服务器要求用户拥有 DBA 权限，才能在运行着的数据库服务器上启动另一个数据库。可以通过 `-gd` 选项将对此选项的访问限定为对已连接到的数据库拥有某一级别权限的用户。允许的值为 DBA、all 或 none。请参见“[-gd 服务器选项](#)”一节第 179 页。
- **创建与删除数据库** 运行个人数据库服务器时，缺省情况下任何用户都可以使用 CREATE DATABASE 语句来创建数据库文件。缺省情况下，网络数据库服务器要求用户拥有 DBA 权限才能创建数据库。可以通过 `-gu` 选项将对此选项的访问限定为对所连接到的数据库拥有某一级别权限的用户。允许的值为 DBA、all、none 或 utility_db。请参见“[-gu 服务器选项](#)”一节第 188 页。
- **停止服务器** `dbstop` 实用程序可以停止数据库服务器。可以在批处理文件中，或者在无法以交互方式停止服务器（通过单击数据库服务器消息窗口上的 [关闭]）的其它情况下使用它。在个人数据库服务器上，缺省情况下任何用户都可以运行 `dbstop` 来关闭服务器。在网络数据库服务器上，缺省设置要求用户拥有 DBA 权限才能停止数据库服务器。可以通过 `-gk` 选项将对此选项的访问限定为对数据库拥有某一级别权限的用户。允许的值为 DBA、all 或 none。请参见“[-gk 服务器选项](#)”一节第 181 页。
- **装载和卸载数据** LOAD TABLE、UNLOAD TABLE 和 UNLOAD 语句都会访问数据库服务器计算机上的文件系统。对于非 Unix 操作系统上的个人数据库服务器，缺省设置为 all；对于网络数据库服务器和 Unix 个人服务器，缺省设置为 DBA。如果运行的是个人数据库服务器，您已经拥有了对文件系统的访问权，这不会成为安全问题。如果运行的是网络数据库服务器，未经授权的文件系统访问可能会成为安全问题。可以通过 `-gl` 选项控制执行数据装载和卸载所需的数据库权限。允许的值为 DBA、all 或 none。请参见“[-gl 服务器选项](#)”一节第 181 页。
- **使用传送层安全对客户端/服务器通信加密** 为提高网络数据包的安全性，可以使用传送层安全来验证客户端应用程序与数据库服务器之间的通信。传送层安全使用椭圆曲线或 RSA 加密技术。请参见“[传送层安全](#)”第 1031 页。
- **禁用数据库功能** `-sf` 服务器选项指定对数据库服务器上运行的数据库禁用的功能列表，这样客户端应用程序或数据库内定义的存储过程、触发器或事件就无法使用这些功能。如果启动的数据库不是自己的，而且该数据库可能包含有害操作（如病毒或特洛伊木马），就可以使用此选项。请参见“[-sf 服务器选项](#)”一节第 207 页。

加密和解密数据库

作为数据库管理员，可以对数据库加密，以增大他人破译数据库中数据的难度。可以选择使用简单加密或高度加密来保护数据库的安全。

注意

如果数据库已加密，则使用 WinZip 等工具对其进行压缩不会使文件明显小于原始数据库文件。

简单加密

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。SQL Anywhere 的早期版本支持简单加密技术。

◆ 使用简单加密

- 使用 `dbinit -ea simple` 选项创建一个数据库。

以下示例使用简单加密创建了数据库 `test.db`：

```
dbinit -ea simple test.db
```

另请参见

- [“初始化实用程序 \(dbinit\)” 一节第 729 页](#)
- [“CREATE DATABASE 语句” 一节 《SQL Anywhere 服务器 - SQL 参考》](#)

高度加密

对数据库使用高度加密技术后，如果没有密钥（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。

小心

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。

支持的高度加密算法

用于实现 SQL Anywhere 高度加密的算法为 AES：它是一种数据块加密算法，美国国家标准与技术协会（National Institute of Standards and Technology，简称 NIST）选择它作为新的数据块编码器高级加密标准（Advanced Encryption Standard，简称 AES）。就性能和大小而言，它具有许多适合于对 SQL Anywhere 数据库进行加密的特性。

也可以使用 AES_FIPS（128 位）或 AES256_FIPS（256 位）类型指定另外一种经 FIPS 认可的 AES 算法来进行高度加密。在以 `-fips` 选项启动数据库服务器时，可以运行用 AES、AES256、AES_FIPS 或 AES256_FIPS 高度加密方法加密的数据库，但不能运行用简单加密方法加密的数据。

库。指定 `-fips` 时，也可以在服务器上启动未加密的数据库。请参见“[-fips 服务器选项](#)”一节第 176 页。

必须在所有用于运行使用 `AES_FIPS` 或 `AES256_FIPS` 加密的数据库的计算机上都安装 SQL Anywhere 安全性组件。

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](#)》。

注意

并非所有平台上都可以使用 FIPS。有关受支持平台的列表，请参见 <http://www.sybase.com/detail?id=1062623>。

控制数据库的高度加密设置

在 SQL Anywhere 中，数据库管理员对高度加密的以下四个方面有控制权：高度加密状态、加密密钥、加密密钥的保护和加密算法。

虽然无法直接在现有数据库中启用或禁用高度加密，但是在实现高度加密时有三种方法可供选择。一种方法是从头开始创建采用高度加密的数据库，另一种方法是重建现有数据库并在重建时更改加密状态，还有一种方法是对现有数据库执行 `CREATE ENCRYPTED DATABASE` 语句。

可以重建数据库以卸载现有数据库的所有数据和模式。这样做会创建一个新数据库（此时可以更改包括高度加密状态在内的各种设置），并将数据重装到新数据库中。需要知道密钥才能卸载高度加密的数据库。

另请参见

- “[重装数据库](#)”一节《[SQL Anywhere 服务器 - SQL 的用法](#)》
- “[初始化实用程序 \(dbinit\)](#)”一节第 729 页
- “[CREATE DATABASE 语句](#)”一节《[SQL Anywhere 服务器 - SQL 参考](#)》
- “[CREATE ENCRYPTED DATABASE 语句](#)”一节《[SQL Anywhere 服务器 - SQL 参考](#)》

创建加密的数据库

要创建加密的数据库，可以采用下列方式：

- 将数据库初始化实用程序 (`dbinit`) 与各种选项配合使用来启用高度加密。
`dbinit` 实用程序 `-ep` 和 `-ek` 选项可以创建采用高度加密的数据库，您可以在提示框或命令行中指定加密密钥。`dbinit -ea` 选项将加密算法设置为 `AES` 或 `AES256` 算法（或 FIPS 认可的 `AES_FIPS` 或 `AES256_FIPS` 算法）。请参见“[初始化实用程序 \(dbinit\)](#)”一节第 729 页。
- 使用 Sybase Central 的 [[创建数据库向导](#)] 创建高度加密的数据库。请参见“[创建数据库 \(Sybase Central\)](#)”一节第 20 页。
- 使用卸载数据库实用程序 (`dbunload`) 并指定相应选项来创建新的采用高度加密的数据库。`-an` 选项创建新的数据库。要在提示框或命令行中指定高度加密和加密密钥，请使用 `-ep` 或 `-ek` 选

项。-ea 选项将加密算法设置为 AES 或 AES256（或 FIPS 认可的 AES_FIPS 或 AES256_FIPS 算法）。请参见“卸载实用程序 (dbunload)”一节第 795 页。

- 您还可以使用 Sybase Central 的 [卸载数据库向导] 创建高度加密的数据库。请参见“使用 [卸载数据库向导] 导出数据”一节《SQL Anywhere 服务器 - SQL 的用法》。
- 使用以下 SQL 语句：
 - “CREATE DATABASE 语句”一节《SQL Anywhere 服务器 - SQL 参考》
 - “CREATE ENCRYPTED DATABASE 语句”一节《SQL Anywhere 服务器 - SQL 参考》
 - “CREATE DECRYPTED FILE 语句”一节《SQL Anywhere 服务器 - SQL 参考》

◆ 创建加密的数据库 (SQL)

1. 从 Interactive SQL 连接到现有数据库。
2. 执行包含 ENCRYPTION 子句、KEY 以及 ALGORITHM 选项的 CREATE DATABASE 语句。

例如，以下语句会使用 FIPS 认可的 128 位 AES 加密在 c:\ 目录中创建一个名为 *myencrypteddb.db* 的数据库文件。

```
CREATE DATABASE 'c:\myencrypteddb.db'
TRANSACTION LOG ON
ENCRYPTED ON
  KEY '0kZ2o52AK#'
  ALGORITHM 'AES_FIPS';
```

◆ 创建加密的数据库（命令提示符）

1. 使用 dbinit 实用程序创建一个数据库。必须附带 -ek 或 -ep 选项，以便相应地在命令提示符处或窗口中指定加密密钥。

以下命令会创建一个高度加密的数据库，并指定加密密钥和算法。

```
dbinit -ek "0kZ2o56AK#" -ea AES_FIPS "myencrypteddb.db"
```

2. 运行以下命令启动数据库：

```
dbeng11 myencrypteddb.db -ek "0kZ2o56AK#"
```

◆ 使用现有数据库创建加密的数据库 (SQL)

1. 从 Interactive SQL 连接到现有数据库（不是您所复制的数据库）。
2. 使用 CREATE ENCRYPTED DATABASE 语句对数据库加密。

以下语句使用数据库文件 *demo.db*，并为该文件创建一个名为 *encryptedDemo.db* 的 AES 加密的副本。

```
CREATE ENCRYPTED DATABASE 'encryptedDemo.db'
FROM 'demo.db'
KEY 'abc'
ALGORITHM 'AES';
```

当执行 CREATE ENCRYPTED DATABASE 语句时，实际上不会加密（覆盖）原文件，而是为该文件创建一个加密形式副本。如果存在与此数据库关联的事务日志、事务日志镜像或 dbspace，则也会创建这些文件的加密副本。请参见“[CREATE ENCRYPTED DATABASE 语句](#)”一节《[SQL Anywhere 服务器 - SQL 参考](#)》。

为技术支持加密数据库

如果有一个需要恢复的数据库，想要对其进行加密以便送给技术支持部门，则必须使用 CREATE ENCRYPTED FILE 语句。任何与数据库相关的文件（例如事务日志、事务日志镜像以及 dbspace 文件）必须也要使用此语句进行加密。请参见“[CREATE ENCRYPTED FILE 语句](#)”一节《[SQL Anywhere 服务器 - SQL 参考](#)》。

比较 CREATE ENCRYPTED DATABASE 和 CREATE ENCRYPTED FILE 语句

当想要对现有的数据库进行加密时，应使用 CREATE ENCRYPTED DATABASE 语句。仅在需要对需要恢复的数据库进行加密时才能使用 CREATE ENCRYPTED FILE 语句。

这两个语句都要求您拥有 DBA 权限，且在执行语句时您不能连接到要加密的数据库。

CREATE ENCRYPTED FILE 和 CREATE ENCRYPTED DATABASE 语句彼此之间的不同点如下：

- CREATE ENCRYPTED FILE 语句必须针对每个与数据库相关的文件（事务日志、事务日志镜像、dbspace，如果有）单独执行，而 CREATE ENCRYPTED DATABASE 语句自动加密所有与数据库相关的文件。
- CREATE ENCRYPTED DATABASE 语句不能在需要恢复的数据库上使用；CREATE ENCRYPTED FILE 语句则可以做到这一点。
- CREATE ENCRYPTED DATABASE 语句不能在过程、触发器或批处理内部使用。CREATE ENCRYPTED FILE 语句则可以做到这一点。
- CREATE ENCRYPTED DATABASE 语句支持 SIMPLE 加密算法，但 CREATE ENCRYPTED FILE 语句则不支持。

另请参见

有关加密密钥的详细信息，请参见“[DatabaseKey 连接参数 \[DBKEY\]](#)”一节第 259 页。

在 Windows Mobile 上，仅 ARM 处理器支持 AES_FIPS 和 AES256_FIPS 算法。

注意

并非所有平台上都可以使用 FIPS。有关受支持平台的列表，请参见 <http://www.sybase.com/detail?id=1062623>。

解密数据库

可以使用 CREATE DECRYPTED DATABASE 语句对数据库进行解密。与使用 CREATE ENCRYPTED DATABASE 语句时的情况一样，实际上是创建该文件的副本（此时是以解密的形式），而不是真的将原数据库文件覆盖。

◆ 解密数据库 (SQL)

1. 从 Interactive SQL 连接到现有数据库。
2. 使用 CREATE DECRYPTED DATABASE 语句对数据库解密。

第一个语句将为 *demo.db* 数据库创建一个 AES256 加密的副本，名为 *demoEncrypted.db*。第二个语句为 *demoEncrypted.db* 创建一个解密副本，名为 *demoDecrypted.db*。

```
CREATE ENCRYPTED DATABASE 'demoEncrypted.db'  
FROM 'demo.db'  
KEY 'Sd8f6654*Mnn'  
ALGORITHM 'AES256';  
CREATE DECRYPTED DATABASE 'demoDecrypted.db'  
FROM 'demoEncrypted.db'  
KEY 'Sd8f6654*Mnn';
```

如果存在与此数据库关联的事务日志、事务日志镜像或 *dbspace*，则也会创建这些文件的解密副本。请参见“[CREATE DECRYPTED DATABASE 语句](#)”一节《[SQL Anywhere 服务器 - SQL 参考](#)》。

为技术支持解密数据库

如果有一个需要恢复的数据库，想要对其进行解密以便送给技术支持部门，则必须使用 CREATE DECRYPTED FILE 语句。任何与数据库相关的文件（例如事务日志、事务日志镜像以及 *dbspace* 文件）必须也要使用此语句进行解密。请参见“[CREATE DECRYPTED FILE 语句](#)”一节《[SQL Anywhere 服务器 - SQL 参考](#)》。

使用加密密钥

最好选择一个无法被轻易猜到的加密密钥值。对密钥的长度没有任何限制，但通常密钥越长越好，因为与较长的密钥相比，较短的密钥更易于猜测。同样，组合使用数字、字母和特殊字符会减少他人猜中密钥的几率。

加密密钥始终区分大小写，它们不能包含前导空格、尾随空格或分号。

每次想要启动数据库时都必须提供此密钥。丢失或忘记密钥会导致数据库完全无法访问。

可以选择是在命令提示符处（缺省设置）还是在提示框中输入加密密钥。选择在提示框中输入密钥增加了安全性，因为这样人们根本无法看清密钥。每次当客户端启动数据库时，都必须指定密钥。在数据库管理员启动数据库的情况下，客户端永远都不需要具有访问密钥的权限。请参见“[-ep 服务器选项](#)”一节第 173 页。

小心

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。

可以使用 CREATE ENCRYPTED DATABASE 语句更改已加密数据库或已启用表加密的数据库的加密密钥。与加密数据库时的情况一样，实际执行的操作并不是覆盖现有文件，而是创建该文件的一个使用新密钥进行加密的副本。

◆ 更改数据库的加密密钥

- 使用 CREATE ENCRYPTED DATABASE 语句更改已加密数据库的加密密钥。

以下示例使用以密钥 abc 加密的数据库文件 *myOldDatabase.db*，并为该文件创建一个使用密钥 abc123 加密的名为 *myNewDatabase.db* 的副本。任何其它与数据库相关的文件（例如事务日志、事务日志镜像以及 dbspace 文件）也使用此新加密密钥进行创建。请参见“CREATE ENCRYPTED DATABASE 语句”一节《SQL Anywhere 服务器 - SQL 参考》。

```
CREATE ENCRYPTED DATABASE myNewDatabase.db
FROM myOldDatabase.db
KEY 'abc123'
OLD KEY 'abc'
ALGORITHM 'AES';
```

性能问题

数据库加密后，SQL Anywhere 的性能会有所下降。对性能的影响取决于从磁盘读取页或向磁盘写入页的频率，可以通过确保服务器使用足够的高速缓存大小来将这种影响降至最低水平。

可以在启动服务器时使用 -c 选项来增加高速缓存的起始大小。对于支持动态调整高速缓存大小的操作系统，所使用的高速缓存大小可能会受到可用内存量的限制；若要增加高速缓存大小，应增加可用内存量。

另请参见

- “使用高速缓存提高性能”一节《SQL Anywhere 服务器 - SQL 的用法》
- “-c 服务器选项”一节第 158 页

加密数据库的某些部分

如果只想要加密数据库的某些部分，则可以选择加密列或表。

列加密可以随时在任意表中的任意列上执行。表加密要求数据库已经启用了表加密。表加密在数据库创建（初始化）时启用。

列加密

如果只想加密数据库中的某些列，可以使用 ENCRYPT 函数来实现。ENCRYPT 函数对传递给它的值进行加密时所用的算法即用于数据库加密的 AES 高度加密算法。

ENCRYPT 函数的密钥区分大小写，即使是在不区分大小写的数据库中也如此。与大多数口令一样，最好选择无法被轻易猜到的密钥值。建议选择满足以下条件的密钥值：长度至少为 16 个字符，混合使用大小写并包含数字、字母和特殊字符。每次要对数据进行解密时，都必须指定此密钥。

小心

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。

加密的值可以使用 DECRYPT 函数进行解密。必须使用 ENCRYPT 函数中指定的那个密钥。这两个函数都会返回 LONG BINARY 值。如果需要其它数据类型的值，可以使用 CAST 函数将值转换为所需的数据类型。下例介绍如何使用 CAST 函数将解密的值转换为所需的数据类型。请参见“CAST 函数 [Data type conversion]”一节《SQL Anywhere 服务器 - SQL 参考》。

如果数据库用户需要访问解密形式的数据，但您不想让他们访问加密密钥，则可以创建一个使用 DECRYPT 函数的视图。这样用户就可以在不知道加密密钥的情况下访问解密数据。创建使用该表的视图或存储过程时，可以使用 ALTER VIEW 和 ALTER PROCEDURE 语句的 SET HIDDEN 参数来确保用户无法通过查看视图或过程定义访问加密密钥。请参见“ALTER PROCEDURE 语句”一节《SQL Anywhere 服务器 - SQL 参考》和“ALTER VIEW 语句”一节《SQL Anywhere 服务器 - SQL 参考》。

列加密示例

以下示例使用触发器对名为 user_info 的表中存储口令的列进行加密。user_info 表的定义如下：

```
CREATE TABLE user_info (
    employee_ID INTEGER NOT NULL PRIMARY KEY,
    user_name CHAR(80),
    user_pwd CHAR(80) );
```

将两个触发器添加到数据库中，以在添加新用户或更新现有用户口令时对 user_pwd 列中的值进行加密。

- 每当在 user_info_table 中添加新行时，都会触发 encrypt_new_user_pwd 触发器：

```
CREATE TRIGGER encrypt_new_user_pwd
BEFORE INSERT
ON user_info
REFERENCING NEW AS new_pwd
FOR EACH ROW
BEGIN
    SET new_pwd.user_pwd=ENCRYPT(new_pwd.user_pwd, '8U3dkA');
END;
```

- 每当在 user_info 表中更新 user_pwd 列时，都会触发 encrypt_updated_pwd 触发器：

```
CREATE TRIGGER encrypt_updated_pwd
BEFORE UPDATE OF user_pwd
ON user_info
REFERENCING NEW AS new_pwd
FOR EACH ROW
BEGIN
    SET new_pwd.user_pwd=ENCRYPT(new_pwd.user_pwd, '8U3dkA');
END;
```

向数据库添加新用户：

```
INSERT INTO user_info
VALUES ( '1', 'd_williamson', 'abc123');
```

如果执行 SELECT 语句以查看 user_info 表中的信息，则可看到 user_pwd 列中的值是二进制数据（口令的加密形式），而不是 INSERT 语句中指定的值 abc123。

如果更改此用户的口令：

```
UPDATE user_info
SET user_pwd='xyz'
WHERE employee_ID='1';
```

将会触发 `encrypt_updated_pwd` 触发器，且 `user_pwd` 列中出现新口令的加密形式。

通过发出以下 SQL 语句可以检索原始口令。此语句使用 `DECRYPT` 函数和加密密钥对数据进行解密，并使用 `CAST` 函数将值从 `LONG BINARY` 转换为 `CHAR` 类型：

```
SELECT CAST (
    DECRYPT( user_pwd, '8U3dkA' )
    AS CHAR(100))
FROM user_info
WHERE employee_ID = '1';
```

另请参见

- “[ENCRYPT 函数 \[String\]](#)” 一节 《SQL Anywhere 服务器 - SQL 参考》
- “[DECRYPT 函数 \[String\]](#)” 一节 《SQL Anywhere 服务器 - SQL 参考》

表加密

表加密可以对包含敏感数据的表或实例化视图进行加密，而且不会造成在对整个数据库加密时可能会导致的性能影响。启用表加密时，会对加密表的表页、相关联的索引页和临时文件页进行加密。还会对包含有关加密表的事务的事务日志页进行加密。

有关对实例化视图进行加密的详细信息，请参见“[加密和解密实例化视图](#)”一节 《SQL Anywhere 服务器 - SQL 的用法》。

要对数据库中的表加密，必须启用表加密。启用表加密必须在数据库初始化时进行。要查看是否启用了表加密，请使用 `DB_PROPERTY` 函数查询 `EncryptionScope` 数据库属性，如下所示：

```
SELECT DB_PROPERTY( 'EncryptionScope' );
```

如果返回值为 `TABLE`，则表明表加密已启用。

要查看表加密当前使用的加密算法，请使用 `DB_PROPERTY` 函数查询 `Encryption` 数据库属性，如下所示：

```
SELECT DB_PROPERTY( 'Encryption' );
```

有关所支持的加密算法列表，请参见“[加密和解密数据库](#)”一节第 1019 页。

表加密对性能的影响

对于加密的表，每个表页在写入磁盘时都会进行加密，在从磁盘读入时都会进行解密。此过程对应用程序不可见。不过，从加密的表读出或向其中写入时，可能会对性能产生轻微的负面影响。对现有表进行加密或解密可能会花费很长的时间，时间长短取决于表的大小。

加密表中列索引的索引页也将被加密，包含有关加密表事务的事务日志页及数据库临时文件中的所有页同样会进行加密。所有其它数据库和事务日志页均未加密。

加密表可能包含压缩列。在这种情况下，数据将先进行压缩，再进行加密。

对表加密不会影响存储要求。

启动启用了表加密的数据库

启动启用了表加密的数据库与启动加密数据库的情况相同。例如，如果启动数据库时使用 `-ek` 选项，则必须指定密钥。如果启动数据库时使用 `-ep` 选项，则会提示您输入密钥。请参见“[初始化实用程序 \(dbinit\)](#)”一节第 729 页。

启用数据库中的表加密

必须在创建数据库时启用和配置表加密。如果数据库未启用表加密，或当前正在执行数据库加密，则必须重新创建数据库以启用表加密。

◆ 创建采用表加密的数据库 (SQL)

- 使用 `CREATE DATABASE` 语句创建数据库，并指定密钥和加密算法。

以下命令会创建为表启用了高度加密的数据库 `new.db`，该加密使用密钥 `abc` 和 `AES256_FIPS` 加密算法：

```
CREATE DATABASE 'new.db'  
  ENCRYPTED TABLE  
  KEY 'abc'  
  ALGORITHM 'AES256_FIPS';
```

以后对此数据库中的表加密时，将使用 `AES256_FIPS` 算法和 `abc` 密钥。

◆ 创建采用表加密的数据库（命令提示符）

- 使用 `dbinit -et` 和 `-ek` 选项创建数据库，并指定密钥和加密算法。

以下命令会创建为表启用了高度加密的数据库 `new.db`，该加密使用密钥 `abc` 和 `AES256_FIPS` 加密算法：

```
dbinit new.db -et -ek abc -ea AES256_FIPS
```

以后对此数据库中的表加密时，将使用 `AES256_FIPS` 算法和 `abc` 密钥。

◆ 使用现有数据库创建采用表加密的数据库 (SQL)

- 使用 `CREATE ENCRYPTED TABLE DATABASE` 语句创建数据库的加密副本，并指定密钥。

下面的示例从一个名为 `contacts1` 的现有数据库创建一个名为 `contacts2` 的数据库。新数据库支持加密表。

```
CREATE ENCRYPTED TABLE DATABASE 'contacts2.db'  
  FROM 'contacts1.db'  
  KEY 'Sd8f6654'  
  OLD KEY 'Sc8e5543';
```

以后对此数据库中的表加密时，将使用 `AES` 算法和 `Sd8f6654` 密钥。

加密表

要对数据库中的表加密，必须已在数据库中启用表加密。请参见“[启用数据库中的表加密](#)”一节第 1027 页。

加密表时，将使用在数据库创建时所指定的加密算法和密钥。

◆ 创建表时对表加密 (SQL)

- 使用 CREATE TABLE 语句的 ENCRYPTED 子句创建表。

以下命令会创建名为 Employees 的加密表：

```
CREATE TABLE MyEmployees (  
    MemberID CHAR(40),  
    CardNumber INTEGER )  
ENCRYPTED;
```

◆ 创建表后对其加密 (SQL)

- 使用 ALTER TABLE 语句的 ENCRYPTED 子句对表加密。

以下语句创建一个名为 MyEmployees2 的表，然后对其加密。

```
CREATE TABLE MyEmployees2 (  
    MemberID CHAR(40),  
    CardNumber INTEGER );  
ALTER TABLE MyEmployees2  
ENCRYPTED;
```

另请参见

- “加密和解密数据库”一节第 1019 页
- “初始化实用程序 (dbinit)”一节第 729 页
- “创建加密的数据库”一节第 1020 页
- “创建数据库 (Sybase Central)”一节第 20 页
- “ALTER TABLE 语句”一节 《SQL Anywhere 服务器 - SQL 参考》
- “CREATE DATABASE 语句”一节 《SQL Anywhere 服务器 - SQL 参考》
- “CREATE ENCRYPTED DATABASE 语句”一节 《SQL Anywhere 服务器 - SQL 参考》
- “CREATE DECRYPTED DATABASE 语句”一节 《SQL Anywhere 服务器 - SQL 参考》
- “CREATE TABLE 语句”一节 《SQL Anywhere 服务器 - SQL 参考》

保护 Windows Mobile 数据库的安全

本节介绍有助于保护 Windows Mobile 数据库安全的 SQL Anywhere 功能。本节将一一详细地介绍审计和数据库加密，概述其它安全功能，并提供可找到详细信息的链接。

Windows 桌面平台的许多 SQL Anywhere 安全功能在 Windows Mobile 上都得到了支持（如数据库文件加密和简单通信加密）或特殊支持（如翻译日志文件实用程序）。

Windows Mobile 上运行的数据库所使用的用户标识和验证功能与 Windows 桌面平台上运行的数据库所使用的相同。这些功能控制可以访问数据库的用户以及这些用户可以执行的操作。请参见“[控制数据库访问](#)”一节第 1006 页。

Windows Mobile 设备安全

如果要在 Windows Mobile 设备上存储敏感数据，可以考虑使用为 Windows Mobile 设备提供的安全功能。

有关可用安全功能的详细信息，请参见随 Windows Mobile 设备提供的《用户手册》。

数据库服务器选项

服务器选项用于控制哪些用户可以在服务器上执行哪些操作。

这些选项是在启动 Windows Mobile 设备上的数据库时，在 **[服务器启动选项]** 窗口的 **[选项]** 字段中设置的。

有关详细信息，请参见“[从命令行控制权限](#)”一节第 45 页。

有关在 Windows Mobile 上设置选项的信息，请参见“[在 Windows Mobile 上指定服务器选项](#)”一节第 329 页。

审计

此功能使用事务日志来维护数据库操作的详细记录。

翻译日志文件实用程序 (dbtran) 用于转换存储在事务日志中的信息，其中包括审计信息。Windows Mobile 不支持 dbtran 实用程序，因此无法转换存储在 Windows Mobile 设备上的日志。要使用此实用程序，请将事务日志文件复制到 PC 上。

有关详细信息，请参见“[审计数据库活动](#)”一节第 1012 页。

Windows Mobile 上的数据库加密

数据库加密功能允许选择数据库加密的级别。可以选择使用简单加密或高度加密来保护数据库的安全。SQL Anywhere 在 Windows Mobile 上同时支持简单加密和高度加密。

简单加密 此级别的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。

SQL Anywhere 的早期版本支持简单加密技术。

高度加密 此级别的加密会将数据库及事务日志文件中包含的信息模糊处理，这样他人单纯通过使用磁盘实用程序查看文件时无法破译这些信息。进行高度加密后，没有密钥的情况下数据库将完全无法访问。在 Windows Mobile 上，仅 ARM 处理器支持 AES_FIPS 和 AES256_FIPS 算法。

有关详细信息，请参见“[加密和解密数据库](#)”一节第 1019 页。

通信加密和 Windows Mobile

在客户端/服务器通信通过网络进行传输的过程中，可以对其进行加密以提高安全性。SQL Anywhere 提供了两种类型的通信加密：简单加密和高度加密。

简单通信加密接受采用简单加密进行加密的通信数据包。所有平台（包括 Windows Mobile 和早期版本的 SQL Anywhere）都支持此级别的通信加密。

Windows Mobile 上不支持高度通信加密。

有关对通信加密的详细信息，请参见“[Encryption 连接参数 \[ENC\]](#)”一节第 265 页。

传送层安全

目录

传送层安全简介	1032
设置传送层安全	1035
创建数字证书	1036
加密 SQL Anywhere 客户端/服务器通信	1042
加密 SQL Anywhere Web 服务	1046
加密 MobiLink 客户端/服务器通信	1047
证书实用程序	1054

传送层安全简介

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》。

传送层安全是一种 IETF 标准协议，它使用数字证书和公共密钥加密算法来保护客户端/服务器的通信安全。传送层安全支持加密、篡改检测和基于证书的验证。

可将传送层安全用于：

- 保护 SQL Anywhere 数据库服务器与客户端应用程序之间的通信安全。
- 保护 MobiLink 服务器和 MobiLink 客户端之间的通信安全。
- 设置安全的 SQL Anywhere Web 服务器。

安全通信始于一些信息的交换（一次握手），包括：

- **服务器验证** 传送层安全使用服务器证书建立和维护安全连接。为每个服务器创建唯一的证书文件。可将服务器验证用于 SQL Anywhere 客户端/服务器通信或 MobiLink 同步：
 - 对于 SQL Anywhere 客户端/服务器通信，数据库客户端将核实 SQL Anywhere 数据库服务器的标识。
 - 对于 MobiLink 同步，MobiLink 客户端（SQL Anywhere 或 UltraLite）将核实 MobiLink 服务器的标识。

效率

传送层安全协议使用公共密钥和对称密钥加密的组合。公共密钥加密提供了更好的验证技术，但却需要大量的计算。一旦建立安全连接，客户端和服务器就会将密钥大小为 128 位的高效对称编码器用于其余通信。

证书

SQL Anywhere 包含一个名为 createcert 的工具，可用于为传送层安全创建 X.509 证书文件。不过，如果需要校验是否存在第三方证书，或者如果需要更多安全证书，则可从证书颁发机构购买。

数据库文件加密

有关数据库文件加密的信息，请参见：

- SQL Anywhere 数据库：“加密和解密数据库”一节第 1019 页
- UltraLite 数据库：“保护 UltraLite 数据库”一节《UltraLite - 数据库管理和参考》

TLS 支持

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》。

此主题详述对 RSA、ECC 和 FIPS 加密的支持。

RSA 加密

RSA 加密随 SQL Anywhere 一起免费提供，并可用于客户端/服务器通信、同步和 Web 服务。免费版未经 FIPS 认证。要执行 FIPS 认证的 RSA 加密，需要单独的许可。

有关支持 RSA 的平台的列表，请参见 <http://www.sybase.com/detail?id=1062623>。

ECC 加密

要执行 ECC 加密，需要单独的许可。

有关支持 ECC 的平台的列表，请参见 <http://www.sybase.com/detail?id=1062623>。

FIPS 认可的加密

FIPS 只能用于 RSA 加密。（ECC 尚未被 FIPS 程序所覆盖。）

FIPS 技术需要一个单独的许可。请参见“单独授权的组件”一节《SQL Anywhere 11 - 简介》。

有关支持 FIPS 的平台的列表，请参见 <http://www.sybase.com/detail?id=1062623>。

FIPS 认可的加密技术

您可以使用经过 FIPS 认可的安全算法加密数据库文件，或加密数据库客户端/服务器通信、Web 服务和 MobiLink 客户端/服务器通信的通信内容。

联邦信息处理标准 (FIPS) 140-2 规定了对安全算法的要求。FIPS 140-2 是由美国和加拿大政府通过美国国家标准与测试协会 (National Institute of Standards and Testing, 简称 NIST) 和加拿大通信安全机构 (Communications Security Establishment, 简称 CSE) 授予的。

SQL Anywhere 使用两个均来自 Certicom 的经 FIPS 认可的模块进行加密。在 Palm OS 上，SQL Anywhere 使用 Certicom Security Builder GSE v1.0.1。其在 <http://csrc.nist.gov/cryptval/140-1/140val-all.htm> 页面上编号为 316。在 Windows (desktop 和 Windows Mobile) 和 Unix 平台上，SQL Anywhere 使用 Certicom Security Builder GSE (FIPS Module v2.0)。它位于同一页面上，编号为 542。

强制使用 FIPS

可以选择通过 FIPS 选项强制使用 FIPS。将 FIPS 选项设置为 on 时，所有安全通信必须通过 FIPS 认可的通道。如果尝试使用非 FIPS RSA，它将自动升级为 FIPS RSA。如果选择了 ECC，则会报告错误 (ECC 不支持 FIPS)。必须为希望强制使用 FIPS 的每台计算机设置 FIPS 选项。SQL Anywhere 和 MobiLink 服务器具有 -fips 命令行选项，客户端具有可使用加密参数进行设置的 fips 选项。

有关使用 FIPS 技术加密 SQL Anywhere 数据库文件的信息，请参见“高度加密”一节第 1019 页。

设置传送层安全

以下步骤概述了设置传送层安全所需的任务。

◆ 设置传送层安全概述

1. 获取数字证书。

您需要标识文件和证书文件。服务器标识文件包含服务器的专用密钥，应和数据库或 MobiLink 服务器安全地保存在一起。将服务器证书文件分发给客户端。

您可以从证书颁发机构购买证书。SQL Anywhere 还提供了创建证书的功能，此功能对于开发和测试十分有用。请参见“[创建数字证书](#)”一节第 1036 页。

2. 如果要为 SQL Anywhere 客户端/服务器应用程序设置传送层安全，请执行以下步骤：

- **启动支持传送层安全的 SQL Anywhere 数据库服务器** 使用 `-ec` 数据库服务器选项指定安全类型、服务器标识文件名以及用以保护服务器专用密钥的口令。

如果还需要通过共享内存允许非加密的连接，请指定 `-es` 选项。

请参见“[启动支持传送层安全的数据库服务器](#)”一节第 1042 页。

- **配置客户端应用程序以使用传送层安全** 使用 Encryption 连接参数 [ENC] 指定受信任证书的路径和文件名。

请参见“[配置客户端应用程序以使用传送层安全](#)”一节第 1043 页。

3. 如果要为 SQL Anywhere Web 服务设置传送层安全，请执行以下步骤：

- **启动支持传送层安全的 SQL Anywhere 数据库服务器** 使用 `-xs` 数据库服务器选项指定安全类型、服务器标识文件名和保护服务器专用密钥的口令。

- **将浏览器或其它 Web 客户端配置为信任证书** 请参见“[加密 SQL Anywhere Web 服务](#)”一节第 1046 页。

4. 如果要为 MobiLink 同步设置传送层安全，请执行以下步骤：

- **启动支持传送层安全的 MobiLink 服务器** 使用 `mlsrv11 -x` 选项指定安全流、服务器标识文件名和用以保护服务器专用密钥的口令。

请参见“[启动支持传送层安全的 MobiLink 服务器](#)”一节第 1047 页。

- **配置 MobiLink 客户端以使用传送层安全** 为 MobiLink 同步客户端实用程序 (dbmlsync) 或 UltraLite 应用程序提供合适的安全或网络协议选项。指定安全流和受信任的服务器证书文件名。

请参见“[配置 MobiLink 客户端以使用传送层安全](#)”一节第 1049 页。

其它入门资源

可在新闻组上发布问题：

- [sybase.public.sqlanywhere.mobilink](#)
- [sybase.public.sqlanywhere.ultralite](#)
- [ianywhere.public.sqlanywhere.qanywhere](#)

创建数字证书

设置传送层安全需要数字证书。您可以从证书颁发机构获取证书，或使用 SQL Anywhere 功能创建证书。

SQL Anywhere 证书创建实用程序

可利用 SQL Anywhere 证书创建实用程序 (createcert) 来使用 RSA 或 ECC 生成 X.509 证书文件。请参见“[证书创建实用程序 \(createcert\)](#)”一节第 705 页。

SQL Anywhere 证书查看器实用程序

可利用 SQL Anywhere 证书查看器实用程序 (viewcert) 来使用 RSA 或 ECC 读取 X.509 证书。请参见“[证书查看器实用程序 \(viewcert\)](#)”一节第 708 页。

用于服务器验证的证书

用于服务器验证的证书文件可以遵循相同的过程来创建。在每种情况下，都要创建标识文件和证书文件。

对于服务器验证，创建服务器标识文件和证书文件，以分发给客户端。

证书配置

证书可以是自签名证书，也可以是由商业或企业证书颁发机构签名的证书。

- **自签名证书** 自签名服务器证书可用于简单设置。请参见“[自签名根证书](#)”一节第 1036 页。
- **企业根证书** 企业根证书可用于签署服务器证书来提高多服务器部署的数据完整性和可扩展性。
 - 您可以将用于签署服务器证书的专用密钥保存在一个安全的中央位置。
 - 对于服务器验证，您无需重新配置客户端即可以添加 MobiLink 或数据库服务器。

请参见“[证书链](#)”一节第 1037 页。

- **商业证书颁发机构** 您可以使用第三方证书颁发机构替代企业根证书。商业证书颁发机构拥有用于保存专用密钥和创建高质量服务器证书的专用设施。

请参见“[证书链](#)”一节第 1037 页和“[全局签名证书](#)”一节第 1039 页。

自签名根证书

自签名根证书可用于只包含一个 MobiLink 或数据库服务器的简单设置。

提示

如果需要多个服务器标识文件，请使用企业级证书链或商业证书颁发机构。证书颁发机构使用专用设施保存根专用密钥，从而实现了可扩展性和更高级别的证书完整性。

有关设置证书链的详细信息，请参见“[证书链](#)”一节第 1037 页。

- **证书** 对于服务器验证证书，自签名证书将分发给客户端。它是一个包含标识信息、服务器的公共密钥和自签名数字签名的电子文档。
- **标识文件** 对于服务器验证证书，标识文件应和 MobiLink 服务器或数据库服务器安全地保存在一起。它是自签名证书（分发给客户端）与相应专用密钥的组合。专用密钥使 MobiLink 或数据库服务器能够解密初始握手中由客户端发送的消息。

另请参见

- “服务器验证”一节第 1049 页
- “启动支持传送层安全的数据库服务器”一节第 1042 页
- “证书创建实用程序 (createcert)”一节第 705 页

证书链

如果需要多个标识文件，您可使用证书链而非自签名证书来提高安全性和可扩展性。证书链需要证书颁发机构或企业根证书来签署标识。

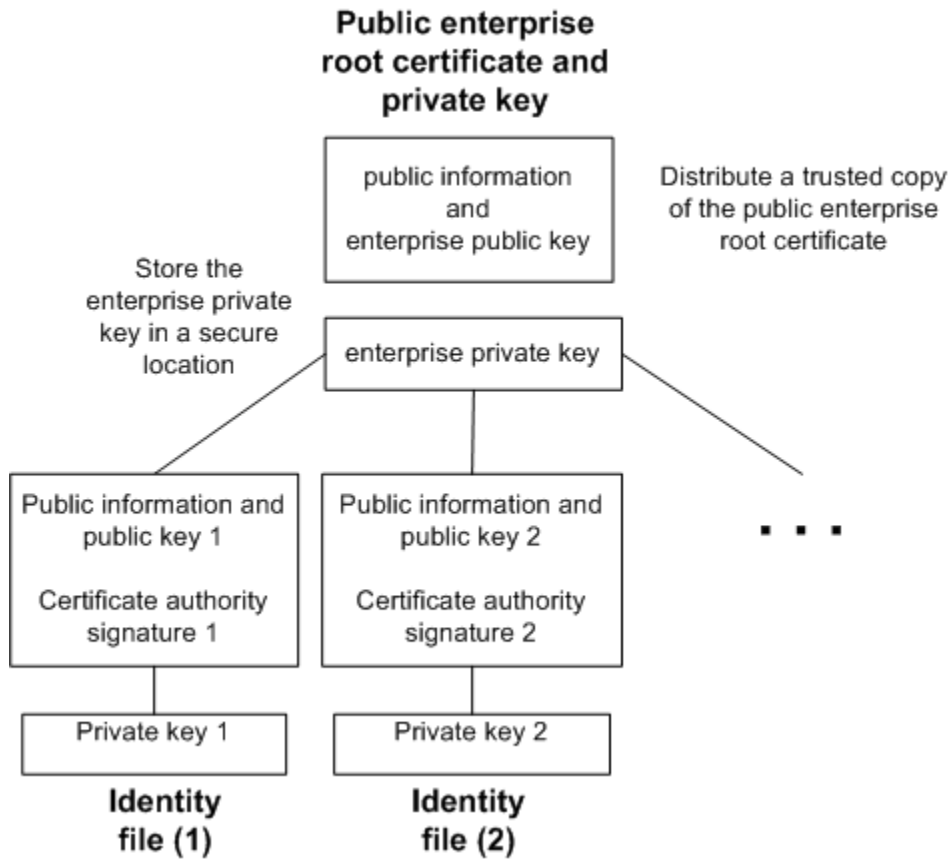
请参见“自签名根证书”一节第 1036 页。

使用证书链的好处

证书链具有以下优点：

- **可扩展性** 对于服务器验证，可将客户端配置为信任由企业根证书或证书颁发机构签名的任何证书。如果添加了一个新的 MobiLink 或数据库服务器，客户端不需要新证书的副本。
- **安全** 企业根证书的专用密钥不存在于标识文件中。请将根证书的专用密钥保存在更安全的位置，或利用能提供专用设施的证书颁发机构，以保护服务器验证的完整性。

下图展示了基本的企业根证书体系结构。



创建用于多服务器环境的证书:

- 生成一个公共企业根证书和企业专用密钥。
将企业专用密钥保存在一个安全位置，最好是专用设施中。
对于服务器验证，公共企业根证书将被分发到客户端。
- 使用企业根证书签署标识。
使用公共企业根证书和企业专用密钥签署每个标识。对于服务器验证，该标识文件用于服务器。

您也可以利用第三方证书颁发机构来签署您的服务器证书。商业证书颁发机构拥有保存专用密钥和创建高质量服务器证书的专用设施。

另请参见

- [“证书创建实用程序 \(createcert\)” 一节第 705 页](#)
- [“全局签名证书” 一节第 1039 页](#)

企业根证书

企业根证书提高了多服务器部署的数据完整性和可扩展性。

- 您可将用于创建受信任证书的专用密钥保存在一个专用设施中。
- 对于服务器验证，您无需重新配置客户端即可添加服务器。

要设置企业根证书，请创建用于签署标识的企业根证书和企业专用密钥。

有关创建服务器证书的信息，请参见“[签名标识文件](#)”一节第 1039 页。

有关生成企业根证书的信息，请参见“[全局签名证书](#)”一节第 1039 页。

签名标识文件

可使用企业根证书签署服务器标识文件。

对于服务器验证，为每个服务器生成标识文件。由于这些证书是由企业根证书签署的，所以请使用 `createcert -s` 选项。

有关生成签名标识文件的信息，请参见“[证书创建实用程序 \(createcert\)](#)”一节第 705 页。

全局签名证书

商业证书颁发机构是从事创建高质量证书并使用这些证书为您的证书请求签名的组织。

全局签名证书有以下优点：

- 在公司间通信的情况下，对公认的外部机构的共同信任将增强对系统安全的信心。证书颁发机构必须保证其签署的所有证书中的标识信息的准确性。
- 证书颁发机构提供生成证书的受控环境和高级方法。
- 根证书的专用密钥必须保持专用。您的组织也许没有合适的位置保存这一至关重要的信息，但证书颁发机构能够设计并维护专用设施，以实现此目的。

设置全局签名证书

要设置全局签名标识文件，您可以执行以下操作：

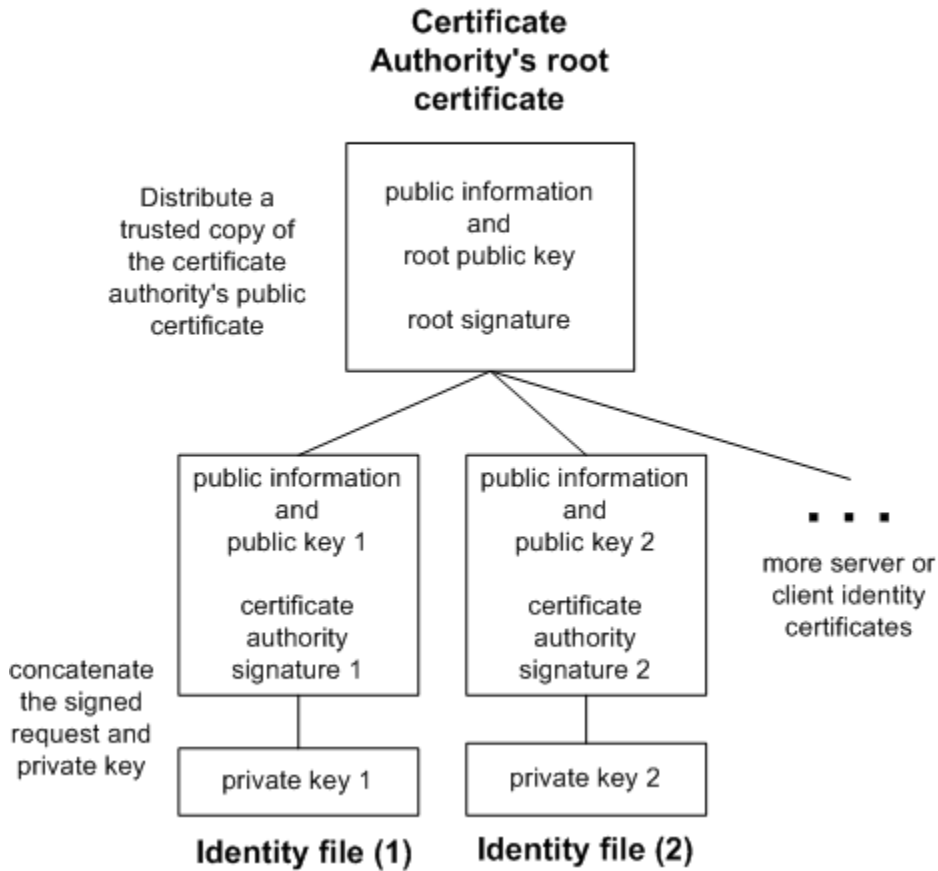
- 使用 `createcert` 实用程序及 `-r` 选项创建证书请求。请参见“[证书创建实用程序 \(createcert\)](#)”一节第 705 页。
- 利用证书颁发机构签署每个请求。可将签名请求与相应的专用密钥相结合来创建服务器标识文件。

对企业根证书进行全局签名

您可以对企业根证书进行全局签名。只有在证书颁发机构生成了可用于对其它证书进行签名的证书的情况下，您才能这么做。

使用全局签名标识文件

您可以直接使用全局签名证书作为服务器标识文件。下图显示了多标识文件的配置：



在 `dbsrv11` 或 `mlsrv11` 命令行中，引用服务器标识文件和专用密钥的口令。

另请参见

- SQL Anywhere: “启动支持传送层安全的数据库服务器”一节第 1042 页
- MobiLink: “启动支持传送层安全的 MobiLink 服务器”一节第 1047 页

将客户端设置为信任证书颁发机构的证书

对于服务器验证，必须确保与联系服务器的客户端信任链中的根证书。对于全局签名证书而言，根证书就是证书颁发机构的证书。

证书字段校验

当使用全局签名证书时，每个客户端必须校验字段值以避免信任的证书已被同一证书颁发机构签署给其它客户端。

请参见“[校验证书字段](#)”一节第 1049 页。

有关将 MobiLink 客户端配置为信任服务器证书的详细信息，请参见“[配置 MobiLink 客户端以使用传送层安全](#)”一节第 1049 页。

有关配置数据库服务器以使用传送层安全的详细信息，请参见“[启动支持传送层安全的数据库服务器](#)”一节第 1042 页。

有关使用全局签名证书建立信任的详细信息，请参见“[全局签名证书](#)”一节第 1039 页。

加密 SQL Anywhere 客户端/服务器通信

可使用传送层安全加密 SQL Anywhere 客户端/服务器通信。

另请参见

- “加密 SQL Anywhere Web 服务” 一节第 1046 页

启动支持传送层安全的数据库服务器

要启动支持传送层安全的数据库服务器，请提供服务器标识文件名和保护服务器专用密钥的口令。

有关设置传送层安全所需步骤的概述，请参见“设置传送层安全”一节第 1035 页。

使用 `-ec` 数据库服务器选项指定 `identity` 和 `identity_password` 参数。如果想通过共享内存允许非加密的连接，还要必须指定 `-es` 选项。

以下是部分 `dbsrv11` 命令行的语法：

```
-ec tls(
  tls_type=cipher;
  identity=server-identity-filename;
  identity_password=password )
-x tcpip
```

- **cipher** 要使用的编码器。对于 RSA 和 ECC 加密，编码器可分别为 **rsa** 或 **ecc**。对于 FIPS 认可的 RSA 加密，请指定 **tls_type=rsa;fips=y**。RSA FIPS 使用单独的认可库，但是与使用 RSA 的 SQL Anywhere 9.0.2 或更高版本的客户端兼容。

有关支持 FIPS 的平台的列表，请参见 <http://www.sybase.com/detail?id=1062623>。

编码器必须与证书创建时指定的加密（ECC 或 RSA）匹配。

有关实施经 FIPS 认可的算法的信息，请参见“**fips 服务器选项**”一节第 176 页。

- **server-identity-filename** 服务器标识文件的路径和文件名。如果使用 FIPS 认可的 RSA 加密，必须使用 RSA 编码器生成证书。

标识文件包含公共证书及其专用密钥。对于不是自签名的证书，标识文件还包含所有签名证书。

有关创建可以自签名或者由证书颁发机构或企业根证书签名的服务器证书的详细信息，请参见“**创建数字证书**”一节第 1036 页。

- **password** 服务器专用密钥的口令。在创建服务器证书时指定此口令。

也可以使用简单加密启动数据库服务器。简单加密使得使用数据包嗅探器读取客户端和服务器之间发送的网络数据包变得更加困难，但不能保证数据的完整性或提供服务器验证。

请参见“**-ec 服务器选项**”一节第 171 页和“**-es 服务器选项**”一节第 174 页。

使用 `-x` 数据库服务器选项指定 TCP/IP 协议。请参见“**-x 服务器选项**”一节第 221 页。

示例

以下示例（所有内容均在一行上输入）使用 `-ec` 数据库服务器选项来指定 ECC 安全、服务器标识文件以及用于保护服务器专用密钥的口令：

```
dbsrv11 -ec tls( tls_type=ecc;identity=c:\test
\srv1_ecc.id;identity_password=mypwd )
-x tcpip c:\test\secure.db
```

您可以使用配置文件和文件隐藏实用程序 (`dbfhide`) 隐藏命令行选项（包括口令）。请参见“[文件隐藏实用程序 \(dbfhide\)](#)”一节第 724 页和“[@data 服务器选项](#)”一节第 156 页。

配置客户端应用程序以使用传送层安全

您可以配置 SQL Anywhere 客户端应用程序以使用传送层安全。使用一组加密连接参数指定受信任证书、加密类型和网络协议。

有关设置传送层安全所需步骤的概述，请参见“[设置传送层安全](#)”一节第 1035 页。

服务器验证

服务器验证允许远程客户端对数据库服务器的标识进行核实。数字签名和证书字段校验应一同使用以实现服务器验证。

数字签名

一个数据库服务器证书包含一个或多个用于保持数据完整性和防止篡改的数字签名。下面是创建数字签名的步骤：

- 一个对证书执行的算法会生成一个唯一值或散列值。
- 此散列值是使用签名证书或证书颁发机构的专用密钥进行加密的。
- 加密的散列值（称为数字签名）嵌入到证书中。

数字签名可以自签名或由企业根证书或证书颁发机构签名。

如果客户端应用程序与数据库服务器联系，并且双方均配置为使用传送层安全，则服务器就会将其证书的副本发送到客户端。客户端使用此证书中包含的服务器公共密钥解密证书的数字签名，计算出证书的一个新散列值，并比较这两个值。如果这两个值匹配，则可以确认服务器证书的完整性。

如果您使用经 FIPS 认可的 RSA 加密，则必须使用 RSA 生成证书。

有关自签名证书的详细信息，请参见“[自签名根证书](#)”一节第 1036 页。

有关企业根证书和证书颁发机构的详细信息，请参见“[证书链](#)”一节第 1037 页。

校证书字段

当使用全局签名证书时，每个客户端必须验证证书字段的值，以避免信任已被同一证书颁发机构签署给其它客户端的证书。这一问题可以通过要求客户端测试证书的标识部分中相应字段的值予以解决。证书颁发机构必须保证其签署的所有证书中的标识信息的准确性。

有关全局签名证书的详细信息，请参见“[全局签名证书](#)”一节第 1039 页。

当使用 `createcert` 实用程序创建证书时，请输入组织、组织单位和公用名字段的值。请使用相应的客户端连接参数校验这些字段。如果您使用第三方证书颁发机构对证书进行全局签名，强烈建议您校证书字段。

- **组织** [组织] 字段对应于 `certificate_company` 加密协议选项。请参见“[certificate_company 协议选项](#)”一节第 287 页。
- **组织单位** [组织单位] 字段对应于 `certificate_unit` 加密协议选项。请参见“[certificate_unit 协议选项](#)”一节第 289 页。
- **公用名** [公用名] 字段对应于 `certificate_name` 加密协议选项。请参见“[certificate_name 协议选项](#)”一节第 288 页。

有关客户端加密连接参数的详细信息，请参见“[Encryption 连接参数 \[ENC\]](#)”一节第 265 页。

使用 `trusted_certificates` 协议选项

如果在 Encryption 连接参数中指定了 TLS，则此选项是唯一必需的协议选项。客户端使用 `trusted_certificates` 加密协议选项指定受信任的数据库服务器证书。受信任证书可以是服务器的自签名证书、公共企业根证书或属于商业证书颁发机构的证书。

另请参见

- “[trusted_certificates 协议选项](#)”一节第 307 页
- “[创建数字证书](#)”一节第 1036 页

建立使用传送层安全的客户端连接

要将客户端应用程序设置为使用传送层安全，请在您的连接字符串中使用 Encryption [ENC] 连接参数。连接字符串采用以下格式（必须将其全部写在一行上）：

```
Encryption=tls(
  tls_type=cipher;
  [ fips={y | n}; ]
  trusted_certificates=public-certificate
  [ certificate_company=organization; ]
  [ certificate_name=common-name; ]
  [ certificate_unit=organization-unit ] )
```

- **cipher** 对于 RSA 和 ECC 加密，可分别为 `rsa` 或 `ecc`。缺省值为 `rsa`。对于 FIPS 认可的 RSA 加密，请指定 `tls_type=rsa;fips=y`。RSA FIPS 使用一个单独的经过认可的库，但是与使用 RSA 的 SQL Anywhere 9.0.2 或更高版本的数据库服务器兼容。不可以用 `tls_type=ecc` 来指定 `fips=y`。

如果编码器与用于创建证书的加密（RSA 或 ECC）不匹配，则连接会失败。

- **public-certificate** 是包含一个或多个受信任证书的文件的路径和文件名。如果您使用经 FIPS 认可的 RSA 加密，则必须使用 RSA 生成证书。请参见“[trusted_certificates 协议选项](#)”一节第 307 页。
- **organization** 只有当证书的 [组织] 字段与此值相匹配时，才强制客户端接受服务器证书。请参见“[certificate_company 协议选项](#)”一节第 287 页。
- **common-name** 只有当证书的 [公用名] 字段与此值相匹配时，才强制客户端接受服务器证书。请参见“[certificate_name 协议选项](#)”一节第 288 页。
- **organization-unit** 只有当证书的 [组织单位] 字段与此值相匹配时，才强制客户端接受服务器证书。请参见“[certificate_unit 协议选项](#)”一节第 289 页。

有关 `trusted_certificates` 及其它客户端安全参数的详细信息，请参见“[校证书字段](#)”一节第 1044 页和“[使用 trusted_certificates 协议选项](#)”一节第 1044 页。

有关创建或获取证书的详细信息，请参见“[创建数字证书](#)”一节第 1036 页。

有关加密连接参数的详细信息，请参见“[Encryption 连接参数 \[ENC\]](#)”一节第 265 页。

示例

以下示例使用 `trusted_certificates` 加密连接参数来指定证书 `public_cert.crt`。

```
"UID=DBA;PWD=sql;ENG=myeng;LINKS=tcPIP;  
ENC=tls(tls_type=ecc;trusted_certificates=public_cert.crt)"
```

以下示例使用 `trusted_certificates` 加密连接参数来指定证书 `public_cert.crt`，并使用 `certificate_unit` 和 `certificate_name` 加密连接参数校证书字段。

```
"UID=DBA;PWD=sql;ENG=myeng;LINKS=tcPIP;  
ENC=tls(tls_type=ecc;trusted_certificates=public_cert.crt;  
certificate_unit=test_unit;certificate_name=my_certificate)"
```

加密 SQL Anywhere Web 服务

SQL Anywhere Web 服务器支持使用 SSL 版本 3.0 和 TLS 版本 1.0 的 HTTPS 连接。

要为 SQL Anywhere Web 服务设置传送层安全，请执行以下步骤：

- **获取数字证书** 您需要数据库服务器证书文件和标识文件。证书（可以是证书颁发机构证书）被分发给浏览器或 Web 客户端。服务器标识文件将安全地存储于您的 SQL Anywhere Web 服务器。

有关创建数字证书的一般信息（包括有关使用证书颁发机构的信息），请参见“[创建数字证书](#)”一节第 1036 页。

- **启动支持传送层安全的 Web 服务器** 使用 `-xs` 数据库服务器选项指定 HTTPS、服务器标识文件和用来保护专用密钥的口令。

以下是部分 `dbsrv11` 命令行的语法。

```
-xs protocol(
  [ fips={ y | n }; ]
  identity=server-identity-filename;
  identity_password=password;... ) ...
```

- **protocol** 可以是 `https`，而对于 FIPS 认可的 RSA 加密，可以是 `fips=y` 的 `https`。FIPS 认可的 HTTPS 使用单独认可的库，但与 HTTPS 兼容。

注意

当使用 FIPS 认可的 HTTPS 时，Mozilla Firefox 浏览器可以连接。然而，多数版本的 Internet Explorer、Opera 或 Safari 浏览器均不支持 FIPS 认可的 HTTPS 所用的编码器套件—如果您要使用 FIPS 认可的 HTTPS，则这些浏览器可能无法连接。

有关实施经 FIPS 认可的算法的信息，请参见“[-fips 服务器选项](#)”一节第 176 页。

- **server-identity-filename** 服务器标识的路径和文件名。对于 HTTPS，必须使用 RSA 证书。

- **password** 服务器专用密钥的口令。在创建服务器证书时指定此口令。

有关 `-xs` 服务器选项的详细信息，请参见“[-xs 服务器选项](#)”一节第 224 页。

有关 `identity` 和 `identity_password` 参数的详细信息，请参见：

- “[Identity 协议选项](#)”一节第 294 页
- “[Identity_Password 协议选项](#)”一节第 294 页

- **配置 Web 客户端** 将浏览器或其它 Web 客户端配置为信任证书。受信任证书可以是自签名的，也可以是企业根证书或证书颁发机构证书。

有关创建数字证书的一般信息（包括有关使用证书颁发机构的信息），请参见“[创建数字证书](#)”一节第 1036 页。

加密 MobiLink 客户端/服务器通信

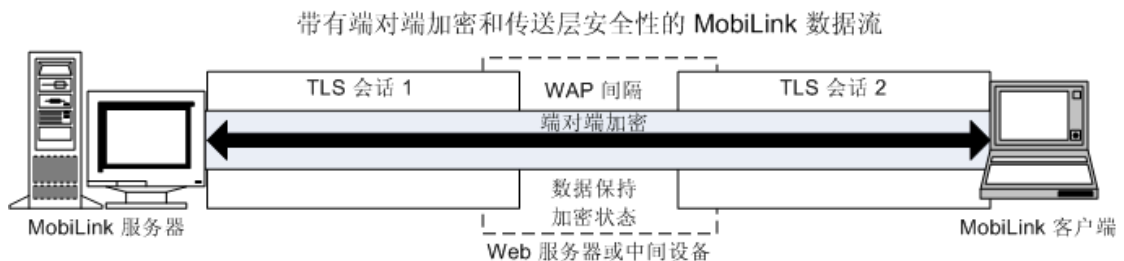
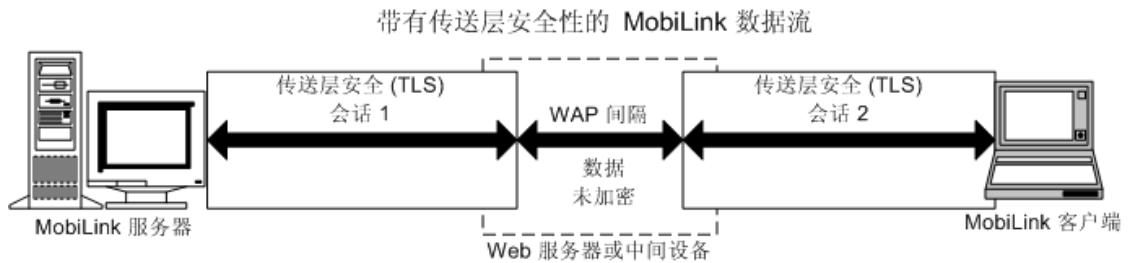
可使用传送层安全加密 MobiLink 客户端/服务器通信。

端对端加密

端对端加密是指在原点加密数据，而在终点解密数据。在传输过程中不会出现数据未加密的情况。

在某些情况下，MobiLink TLS 仅用于加密由客户端和服务器之间的中间设备（例如，加密/解密硬件）所负责的数据。在中间设备中，中间设备先解密数据然后再次对其进行加密，以完成余下的传输过程。特别是，当使用 HTTPS 通过 Web 服务器进行同步时将发生上述情况。数据在中间设备中处于未加密状态的短暂间隔有时被称作无线应用程序协议间隔或 WAP 间隔 (Wireless Application Protocol gap)。

在公司内部，中间设备一般由公司所控制，因此 WAP 间隔通常可以接受。但是，在第三方托管环境中，来自不同公司的数据都将经过相同的 WAP 间隔，因而敏感数据有可能被公开。端对端加密会阻止任何中间设备访问这些数据，这是因为同步流自始至终都处于加密状态，并且还可以选择使用 TLS 再次对其进行加密。



启动支持传送层安全的 MobiLink 服务器

若要启动支持传送层安全的 MobiLink 服务器，请提供标识文件和用于保护服务器专用密钥的标识口令。

有关设置传送层安全所需步骤的概述，请参见“[设置传送层安全](#)”一节第 1035 页。

通过 TCP/IP 和 HTTPS 保护 MobiLink 服务器的安全

使用 `mlsrv11 -x` 服务器选项指定标识和标识口令。以下是 `mlsrv11` 命令行的一部分（必须写在一行上）：

```
-x protocol(
  tls_type=cipher;
  fips={ y | n };
  identity=identity-file;
  identity_password=password;... )
```

- **protocol** 要使用的协议。可以是 **https** 或 **tls**。**tls** 是支持 TLS 的 TCP/IP 协议。
- **cipher** 要使用的编码器。对于 RSA 和 ECC 加密，可分别为 **rsa** 或 **ecc**。编码器必须与用于创建标识的加密匹配。
- **fips** 指示是否使用 FIPS。FIPS 只能与 RSA 加密一起使用。RSA FIPS 使用 Certicom 的单独的通过 FIPS 140-2 认证的软件。使用 FIPS 的服务器与不使用 FIPS 的客户端兼容，反之亦然。RSA FIPS 可用于任何受支持的 32 位 Windows 平台或 Solaris 上的 SQL Anywhere 客户端，或用于 Unix 或任何受支持的 32 位 Windows 平台（包括 Windows Mobile）上的 UltraLite 客户端。
- **identity-file** 标识文件的路径和文件名，其中包含服务器的专用密钥和服务器证书，有时也包含证书颁发机构签名的证书。
有关创建可以自签名或者由证书颁发机构或企业根证书签名的服务器证书的信息，请参见“[创建数字证书](#)”一节第 1036 页。
- **password** 服务器专用密钥的口令。在创建服务器标识时指定此口令。

请参见“[-x 选项](#)”一节《[MobiLink - 服务器管理](#)》。

示例

以下示例在 `mlsrv11` 命令行上指定安全类型 (RSA)、服务器标识文件及用以保护服务器专用密钥的标识口令：

```
mlsrv11 -c "dsn=my_cons"
-x tls(tls_type=rsa;identity=c:\test\serv_rsa1.crt;identity_password=pwd)
```

以下示例在 `mlsrv11` 命令行上指定了 ECC 标识：

```
mlsrv11 -c "dsn=my_cons"
-x tls(tls_type=ecc;identity=c:\test\serv_ecc1.crt;identity_password=pwd)
```

以下示例与前面的示例相似，不同之处在于本例的标识文件名中包含空格：

```
mlsrv11 -c "dsn=my_cons"
-x "tls(tls_type=rsa;identity=c:\Program Files\test
\serv_rsa1.crt;identity_password=pwd)"
```

有关 `mlsrv11 -x` 选项的详细信息，请参见“[-x 选项](#)”一节《[MobiLink - 服务器管理](#)》。

有关创建服务器标识文件（在本例中为 `serv_ecc1.crt`）的详细信息，请参见“[创建数字证书](#)”一节第 1036 页。

您可以使用配置文件和文件隐藏实用程序 (dbfhide) 隐藏命令行选项。请参见“@data 选项”一节《MobiLink - 服务器管理》。

配置 MobiLink 客户端以使用传送层安全

您可以配置 SQL Anywhere 或 UltraLite 客户端以使用 MobiLink 传送层安全。对于每个客户端，请指定受信任证书、加密类型和网络协议。

有关设置传送层安全所需步骤的概述，请参见“设置传送层安全”一节第 1035 页。

服务器验证

服务器验证允许远程客户端对服务器的标识进行核实。数字签名和证书字段校验应一同使用以实现服务器验证。

数字签名

服务器证书包含一个或多个用于保持数据完整性和防止篡改的数字签名。下面是创建数字签名的步骤：

- 一个对证书执行的算法会生成一个唯一值或散列值。
- 此散列值是使用签名证书或证书颁发机构的专用密钥进行加密的。
- 加密的散列值（称为数字签名）嵌入到证书中。

数字签名可以自签名或由企业根证书或证书颁发机构签名。

如果 MobiLink 客户端与 MobiLink 服务器联系，并且双方均配置为使用传送层安全，则服务器会将其证书的副本发送到客户端。客户端使用此证书中包含的服务器公共密钥解密证书的数字签名，计算出证书的一个新散列值，并比较这两个值。如果这两个值匹配，则可以确认服务器证书的完整性。

有关自签名证书的详细信息，请参见“自签名根证书”一节第 1036 页。

有关企业根证书和证书颁发机构的详细信息，请参见“证书链”一节第 1037 页。

校证书字段

当使用全局签名证书时，每个客户端必须验证证书字段的值，以避免信任已被同一证书颁发机构签署给其它客户端的证书。这一问题可以通过要求客户端测试证书的标识部分中相应字段的值予以解决。证书颁发机构必须保证其签署的所有证书中的标识信息的准确性。

有关全局签名证书的详细信息，请参见“全局签名证书”一节第 1039 页。

当使用 createcert 实用程序创建证书时，请输入组织、组织单位和公用名字段的值。使用相应的 MobiLink 客户端连接参数可以校验这些字段。

- **组织** 组织字段对应于 `certificate_company` MobiLink 客户端连接参数。请参见“[certificate_company](#)”一节《MobiLink - 客户端管理》。
- **组织单位** 组织单位字段对应于 `certificate_unit` MobiLink 客户端连接参数。请参见“[certificate_unit](#)”一节《MobiLink - 客户端管理》。
- **公用名** 公用名字段对应于 `certificate_name` MobiLink 客户端连接参数。请参见“[certificate_name](#)”一节《MobiLink - 客户端管理》。

有关设置 MobiLink 客户端的详细信息，请参见：

- “[配置 UltraLite 客户端以使用传送层安全](#)”一节第 1052 页
- “[客户端安全选项](#)”一节第 1050 页

有关创建数字证书的详细信息，请参见“[创建数字证书](#)”一节第 1036 页。

客户端安全选项

MobiLink 客户端（SQL Anywhere 和 UltraLite）使用一组公用连接参数配置传送层安全。

trusted_certificates 协议选项

MobiLink 客户端使用 `trusted_certificates` 协议选项来指定受信任的 MobiLink 服务器证书。受信任证书可以是服务器的自签名证书、公共企业根证书或属于商业证书颁发机构的证书。

请参见：

- “[trusted_certificates](#)”一节《MobiLink - 客户端管理》
- “[创建数字证书](#)”一节第 1036 页

校证书字段

`certificate_company`、`certificate_unit` 和 `certificate_name` 协议选项用于核实证书字段，这是服务器验证的一个重要步骤。如果您使用第三方证书颁发机构对证书进行全局签名，强烈建议您校证书字段。

请参见：

- “[校证书字段](#)”一节第 1049 页
- “[全局签名证书](#)”一节第 1039 页
- “[服务器验证](#)”一节第 1049 页

配置 SQL Anywhere 客户端以使用传送层安全

本节介绍如何配置 SQL Anywhere 客户端以使用 HTTPS 或 TCP/IP 传送层安全。

使用 TCP/IP 和 HTTPS 传送层安全

MobiLink 传送层安全是 MobiLink HTTPS 和 TCP/IP 协议的固有功能。要使用 HTTPS 传送层安全，请使用 ADR 扩展选项指定 `trusted_certificates` 连接参数。以下是部分 `dbmlsync` 命令行的语法。

```
-e "ctp=protocol;
adr=[ fips={ y | n }; ]
trusted_certificates=public-certificate;
..."
```

- **protocol** 要使用的协议。可以是 **https** 或 **tls**。**tls** 是使用传送层安全的 TCP/IP 协议。
- **fips** 指示是否使用 FIPS。FIPS 只能与 RSA 加密一起使用。FIPS 认可的 HTTPS 使用 Certicom 的单独的通过 FIPS 140-2 认证的软件，但是它与使用 HTTPS 的 9.0.2 版本或更高版本的 MobiLink 服务器兼容。
- **public-certificate** 受信任证书的路径和文件名。

对于 HTTPS 或 FIPS 认可的 HTTPS，必须使用采用 RSA 加密创建的证书。

另请参见

- “客户端安全选项”一节第 1050 页
- “创建数字证书”一节第 1036 页
- “CommunicationAddress (adr) 扩展选项”一节 《MobiLink - 客户端管理》
- “MobiLink 客户端网络协议选项汇总”一节 《MobiLink - 客户端管理》
- “CREATE SYNCHRONIZATION SUBSCRIPTION 语句 [MobiLink]”一节 《SQL Anywhere 服务器 - SQL 参考》
- “ALTER SYNCHRONIZATION SUBSCRIPTION 语句 [MobiLink]”一节 《SQL Anywhere 服务器 - SQL 参考》

示例

以下示例指定了采用 HTTPS 的 RSA 安全性。该命令必须全部写在一行中：

```
dbmlsync -c "eng=reml;uid=dba;pwd=myspwd"
-e "ctp=https;
adr='trusted_certificates=c:\temp\public_cert.crt;
certificate_company=Sybase, Inc.;
certificate_unit=IAS;
certificate_name=MobiLink'"
```

或者，您可以使用 `CREATE SYNCHRONIZATION SUBSCRIPTION` 或 `ALTER SYNCHRONIZATION SUBSCRIPTION` 语句指定 `CommunicationAddress` 扩展选项。此方法提供同样的信息，但是将信息存储在数据库中。

```
CREATE SYNCHRONIZATION SUBSCRIPTION
TO publ
FOR user1
ADDRESS 'trusted_certificates=c:\temp\public_cert.crt;
certificate_company=Sybase, Inc.;
certificate_unit=IAS;
certificate_name=MobiLink';
```

以下示例指定使用 RSA 安全和 TCP/IP。该命令必须全部写在一行中：

```
dbmlsync -c "eng=reml;uid=myuid;pwd=mypwd"
-e "ctp=tls;
    adr='port=3333;
        tls_type=rsa;
        trusted_certificates=c:\test\public_cert.crt;
        certificate_company=Sybase, Inc.;
        certificate_unit=IAS;
        certificate_name=MobiLink'"
```

或者，您可以使用 CREATE SYNCHRONIZATION SUBSCRIPTION 或 ALTER SYNCHRONIZATION SUBSCRIPTION 语句指定 CommunicationAddress 扩展选项：

```
CREATE SYNCHRONIZATION SUBSCRIPTION
TO publ
FOR user1
ADDRESS 'port=3333;
        tls_type=rsa;trusted_certificates=public_cert.crt;
        certificate_company=Sybase, Inc.;
        certificate_unit=IAS;
        certificate_name=MobiLink';
```

配置 UltraLite 客户端以使用传送层安全

MobiLink 传送层安全是 MobiLink HTTPS 协议的固有功能。如果您使用 HTTPS 和 UltraLite 客户端，则可以直接将受信任证书和证书字段指定为网络协议选项。

有关为 UltraLite 接口指定 HTTPS 协议的详细信息，请参见“[UltraLite 同步流的网络协议选项](#)”一节《[UltraLite - 数据库管理和参考](#)》。

有关 tls_type 同步参数的详细信息，请参见“[tls_type](#)”一节《[MobiLink - 客户端管理](#)》。

◆ 配置 UltraLite 客户端以使用 TCP/IP 或 HTTPS 传送层安全

1. 有两种方法可指定受信任的根证书：

- **创建 UltraLite 数据库时** 请参见“[UltraLite 创建数据库实用程序 \(ulcreate\)](#)”一节《[UltraLite - 数据库管理和参考](#)》或“[UltraLite 初始化数据库实用程序 \(ulinit\)](#)”一节《[UltraLite - 数据库管理和参考](#)》。
- **使用 trusted_certificates 协议选项** 有关详细信息，请参见此过程中的步骤 3。此选项在 Palm OS 上不可用。

2. 为同步指定 TCP/IP 或 HTTPS 协议。安全 TCP/IP 的关键字为 tls。

以下示例是用 C/C++ UltraLite 编写的。要指定 tls，请将 https 更改为 tls。

```
auto ul_synch_info synch_info;
conn.InitSynchInfo( &synch_info );
synch_info.user_name = UL_TEXT( "50" );
synch_info.version = UL_TEXT( "ul_default" );
...
synch_info.stream = "https";
...
```

3. 指定 TCP/IP 或 HTTPS 协议选项。

以下示例是用 C/C++ UltraLite 编写的。要指定 tls，请将 https 更改为 tls。

```
auto ul_synch_info synch_info;
...
synch_info.stream = "https";
synch_info.stream_parms = TEXT(
    "port=9999;
    certificate_company=Sybase, Inc.;
    certificate_unit=IAS;
    certificate_name=MobiLink");
```

certificate_company、certificate_unit 和 certificate_name 协议选项用于校证书字。

请参见“[校证书字](#)”一节第 1049 页。

您还可以指定 trusted_certificates HTTPS 协议选项，它将替代 UltraLite 数据库中所嵌入的任何受信任证书信息（此过程的第 1 步）。trusted_certificates 协议选项在 Palm OS 上不可用。

```
auto ul_synch_info synch_info;
...
synch_info.stream = "https";
synch_info.stream_parms = TEXT(
    "port=9999;
    trusted_certificates=\rsaroot.crt;
    certificate_company=Sybase, Inc.;
    certificate_unit=IAS;
    certificate_name=MobiLink");
```

有关 HTTPS 选项的详细信息，请参见“[UltraLite 同步流的网络协议选项](#)”一节《[UltraLite - 数据库管理和参考](#)》。

证书实用程序

用户通常会转到第三方购买证书。这些证书颁发机构提供其自己用于创建证书的工具。以下工具会对创建用于开发和测试目的的证书十分有用，但也可以用于创建生产证书。

请参见：

- “证书创建实用程序 ([createcert](#))” 一节第 705 页
- “证书查看器实用程序 ([viewcert](#))” 一节第 708 页

复制

本节介绍如何将 SQL Anywhere 用作 Open Server，以及如何使用复制服务器来复制数据。

将 SQL Anywhere 用作 Open Server	1057
使用复制服务器复制数据	1069

将 SQL Anywhere 用作 Open Server

目录

Open Client、Open Server 和 TDS	1058
将 SQL Anywhere 设置为 Open Server	1060
配置 Open Server	1062
Open Client 和 jConnect 连接的特性	1067

Open Client、Open Server 和 TDS

SQL Anywhere 可以作为客户端应用程序的 Open Server 出现。该功能使 Sybase Open Client 应用程序能够以本地方式连接到 SQL Anywhere 数据库。

如果只是希望将某个 Sybase 应用程序与 SQL Anywhere 配合使用，并不需要了解 Open Client、Open Server 或 TDS 的任何详细信息。但是，了解这些组件如何协同工作可能会有助于您配置数据库和设置应用程序。本节说明这些组件如何协同工作，但不对这些组件的内部功能做任何阐述。

Open Client 和 Open Server

SQL Anywhere 和 Adaptive Server 系列产品的其它成员用作 **Open Server**。这意味着您可以使用 Sybase 中可用的 **Open Client** 库来开发客户端应用程序。Open Client 包括客户端库 (CT-Library) 和旧版本的 DB-Library 接口。

有关开发配合 SQL Anywhere 使用的 Open Client 应用程序的信息，请参见“[Sybase Open Client API](#)”《[SQL Anywhere 服务器 - 编程](#)》。

表格式数据流

Open Client 和 Open Server 使用一种称为**表格式数据流** (Tabular Data Stream, 简称 TDS) 的应用程序协议来交换信息。所有使用 Sybase Open Client 库构建的应用程序也都是 TDS 应用程序，因为由 Open Client 库来处理 TDS 接口。不过，某些应用程序（如 jConnect）也属于 TDS 应用程序，尽管它们并不使用 Sybase Open Client 库—它们使用 TDS 协议直接进行通信。

尽管许多 Open Server 使用 Sybase Open Server 库来处理与 TDS 的接口，但某些应用程序拥有自己的与 TDS 的直接接口。Sybase Adaptive Server Enterprise 和 SQL Anywhere 都有内部 TDS 接口。它们作为客户端应用程序的 Open Server 出现，但不使用 Sybase Open Server 库。

编程接口和应用程序协议

SQL Anywhere 支持两种应用程序协议。Open Client 应用程序和其它 Sybase 应用程序（例如，复制服务器和 OmniConnect）使用 TDS。ODBC 和嵌入式 SQL 应用程序使用单独的特定于 SQL Anywhere 的应用程序协议。

TDS 使用 TCP/IP

TDS 之类的应用程序协议建立于处理网络通信的较低级通信协议之上。SQL Anywhere 仅支持 TCP/IP 网络协议上的 TDS。与之相比，特定于 SQL Anywhere 的应用程序协议支持数种网络协议，以及专用于同一台计算机通信的共享内存协议。

Sybase 应用程序和 SQL Anywhere

SQL Anywhere 可以用作 Open Server，这使 Sybase 应用程序（如复制服务器和 OmniConnect）能够与 SQL Anywhere 协同工作。

复制服务器支持

Open Server 接口实现了对 Sybase 复制服务器的支持：复制服务器通过 Open Server 接口连接，从而使 SQL Anywhere 数据库可以作为复制服务器安装中的复制站点。

若要将数据库作为复制服务器安装中的主站点，还要必须使用 Sybase SQL Anywhere 复制代理，也称为**日志传送管理器**。

有关复制代理的信息，请参见“[使用复制服务器复制数据](#)”第 1069 页。

OmniConnect 支持

Sybase OmniConnect 提供了组织内异构数据的统一视图，使用户不需要事先了解数据的特征或查找位置，便可访问多个数据源。此外，OmniConnect 还会对整个企业的数据库执行异构连接，从而实现目标（如 DB2、Sybase Adaptive Server Enterprise、Oracle 和 VSAM）的跨平台表连接。

通过使用 Open Server 接口，可以将 SQL Anywhere 作为 OmniConnect 的数据源。

将 SQL Anywhere 设置为 Open Server

本节介绍如何设置 SQL Anywhere 服务器，使其可以接收来自 Open Client 应用程序的连接。

系统要求

客户端和服务端对于使用 SQL Anywhere 作为 Open Server 有着不同的要求。

服务器端要求

要将 SQL Anywhere 用作 Open Server，服务器端必须具有以下组件：

- **SQL Anywhere 服务器组件** 如果要通过网络访问 Open Server，则必须使用网络服务器 (*dbsrv11.exe*)。仅对于来自同一计算机的连接，才可将个人服务器 (*dbeng11.exe*) 用作 Open Server。
- **TCP/IP** 即使不是通过网络进行连接，也必须具有 TCP/IP 协议栈，才能将 SQL Anywhere 用作 Open Server。

客户端要求

要使用 Sybase 客户端应用程序连接到 Open Server（包括 SQL Anywhere），需要安装以下组件：

- **Open Client 组件** 在您的应用程序使用 Open Client 的情况下，Open Client 库为您的应用程序提供了通过 TDS 进行通信所需的网络库。
- **jConnect** 如果应用程序使用 JDBC，则需要 jConnect 和 Java 运行时环境。SQL Anywhere 支持 jConnect 5.5 和 6.0.5，两者都可从以下网址获得：<http://www.sybase.com/products/informationmanagement/softwaredeveloperkit/jconnect>。
- **DSEdit** 需要 DSEdit（目录服务编辑器）来使服务器名可供 Open Client 应用程序使用。在 Unix 平台上，该实用程序称为 *sybinit*。

DSEdit 不随 SQL Anywhere 提供，但随 Open Server 软件提供。

将数据库服务器作为 Open Server 启动

如果要将 SQL Anywhere 用作 Open Server，必须确保使用 TCP/IP 协议来启动它。缺省情况下，服务器启动所有可用的通信协议，但您可以通过在命令中显式列出这些协议来限制启动的协议。例如，以下两个命令都有效：

```
dbsrv11 -x tcpip -n myserver c:\mydata.db
```

在同一台计算机上进行通信时，可以使用个人数据库服务器作为 Open Server，因为它支持 TCP/IP 协议。

服务器可以在通过 TDS 为 Open Client 应用程序提供服务的同时，使用特定于 SQL Anywhere 的应用程序协议，通过 TCP/IP 协议或其它协议为其它应用程序提供服务。

端口号

计算机上使用 TCP/IP 的每个应用程序都使用不同的 TCP/IP 端口，以便让网络数据包能够最终到达正确的应用程序。SQL Anywhere 的缺省端口为 2638。建议使用该缺省端口号，因为 Internet 编号授权委员会（Internet Assigned Numbers Authority，简称 IANA）已准许 SQL Anywhere 使用该端口号。如果要使用其它端口号，可以使用 ServerPort (PORT) 协议选项指定所用的端口号：

```
dbsrv11 -x tcpip(ServerPort=2629) -n myserver c:\mydata.db
```

如果有多个本地数据库服务器在运行，或要连接到网络服务器，则可能还需要提供一个 ServerName。

Open Client 设置

要连接到某个服务器，客户端计算机的接口文件必须包含一个条目，该条目指定该数据库服务器运行时所在计算机的名称及其使用的 TCP/IP 端口。

有关设置客户端计算机的信息，请参见“[配置 Open Server](#)”一节第 1062 页。

配置 Open Server

SQL Anywhere 可以与网络上的其它 Adaptive Server、Open Server 应用程序及客户端软件进行通信。客户端可以与一个或多个服务器对话，而服务器可以通过远程过程调用与其它服务器进行通信。彼此进行交互的产品需要知道对方在网络上的驻留位置。此网络服务信息存储在接口文件中。

接口文件

接口文件在 Windows 操作系统上通常命名为 *SQL.ini*，在 Unix 操作系统上通常命名为 *interfaces* 或 *interfac*。

接口文件如同一个地址簿，它列出了计算机上 Open Client 应用程序已知的所有数据库服务器的名称和地址。当使用 Open Client 程序连接数据库服务器时，该程序会在接口文件中查找服务器名，然后连接到使用该地址的服务器。

操作系统不同，接口文件的名称、位置和内容也不相同。此外，接口文件中的地址格式也会因网络协议的不同而有差异。

安装 SQL Anywhere 时，安装程序会创建一个简单的接口文件，可以使用该文件通过 TCP/IP 进行与 SQL Anywhere 的本地连接。系统管理员负责修改接口文件，并将其分发给用户，以便用户可以通过网络连接到 SQL Anywhere。

使用 DSEdit 实用程序

DSEdit 实用程序是一个 Windows 实用程序，可以通过它来配置接口文件 (*SQL.ini*)。以下几节说明如何使用 DSEdit 实用程序来配置接口文件。

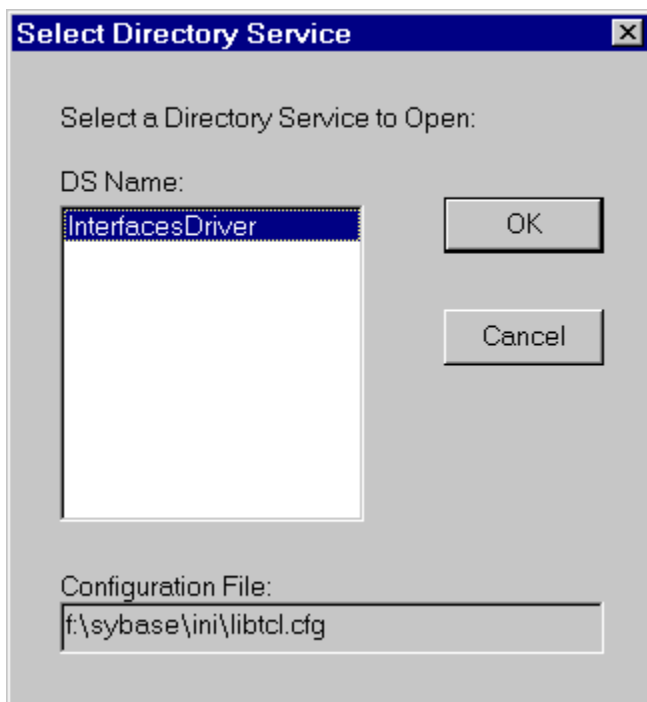
这几节会介绍如何使用 DSEdit 来执行 SQL Anywhere 所需的那些任务。这些不是 DSEdit 实用程序的全部文档。

有关 DSEdit 的详细信息，请参见所用平台的 *配置指南*（随其它 Sybase 产品提供）。

启动 DSEdit

DSEdit 可执行文件位于 *SYBASE\bin* 目录中，该目录会在安装时添加到路径中。

启动 DSEdit 时，会出现 **[Select Directory Service]** 窗口。



打开目录服务会话

[**Select Directory Service**] 窗口允许您打开一个目录服务的会话。通过打开会话，您可以编辑接口文件 (*SQL.ini*)，或编辑任何在 *libtcl.cfg* 文件中列有驱动程序的目录服务。

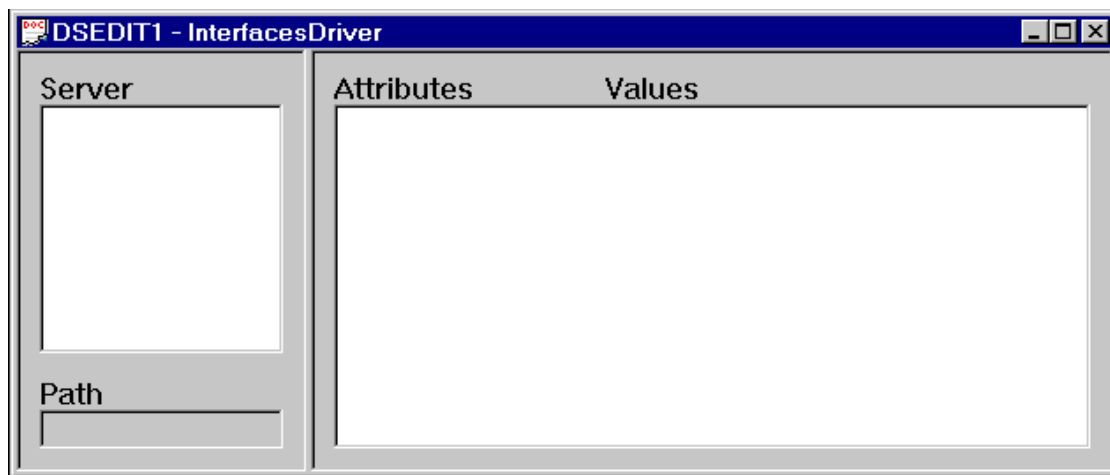
◆ 打开会话：

- 在 [DS Name] 列表中，单击要连接的目录服务的本地名称，然后单击 [OK]。

如果是 SQL Anywhere，请选择 [**InterfacesDriver**]。

必须设置 SYBASE 环境变量

DSEdit 实用程序使用 SYBASE 环境变量来定位 *libtcl.cfg* 文件。如果 SYBASE 环境变量不正确，DSEdit 将无法定位 *libtcl.cfg* 文件。



可以在 [InterfacesDriver] 窗口中添加、修改或删除服务器（包括 SQL Anywhere 服务器）的条目。

添加服务器条目

◆ 添加服务器条目：

1. 从 [Server Object] 菜单选择 [Add]。
2. 在 [Server Name] 框中键入服务器名并单击 [OK]。

该服务器名条目必须与计划连接到的数据库名相符。服务器地址用于标识和定位服务器。
[Server Name] 字段是为 Open Client 提供的标识符。对于 SQL Anywhere，如果数据库服务器装载了多个数据库，DSEdit 服务器名条目会标识要使用的数据库。

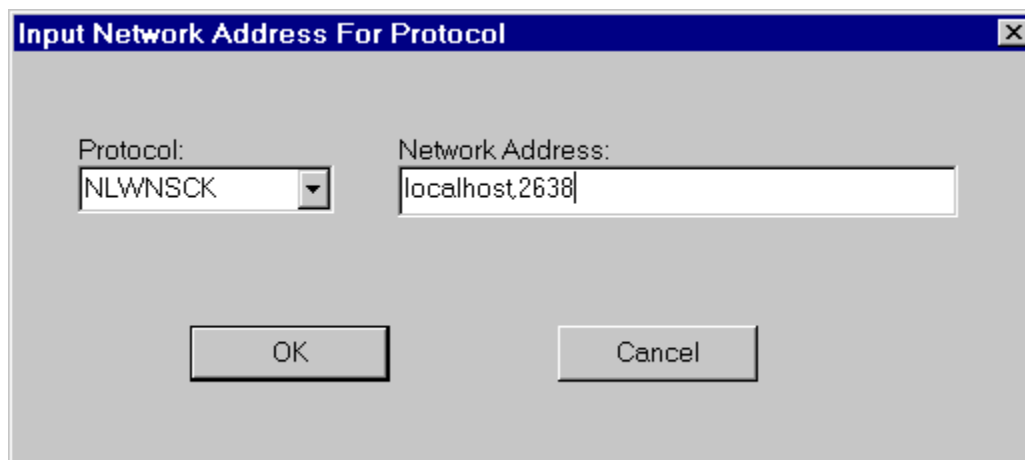
该服务器条目会出现在 [Server] 框中。要指定服务器的属性，必须修改该条目。

添加或更改服务器地址

输入 [Server Name] 后，需要修改 [Server Address] 以完成接口文件条目。

◆ 输入服务器地址

1. 在 [Server] 框中选择服务器条目。
2. 在 [Attributes] 框中右击服务器地址并选择 [Modify Attribute]。
3. 单击 [Add]。
4. 在 [Protocol] 列表中选择 [NLWNSCK]（此为 TCP/IP 协议）。



5. 在 [Network Address] 字段中键入有效的网络地址。对于 TCP/IP 地址，使用下列两种格式之一：

- 计算机名称, 端口号
- IP 地址, 端口号

地址或计算机名与端口号之间用逗号分隔。

计算机名 用于标识运行服务器的计算机的名称（或 IP 地址）。在 Windows 操作系统中，可以在 [控制面板] 的 [网络设置] 中找到计算机名。

即使客户端和服务端位于同一台计算机上，您仍然必须输入计算机名。此时，您可以使用 **localhost** 来标识当前计算机。

端口号 所输入的端口号必须与启动 SQL Anywhere 数据库服务器时所使用的端口号相符。请参见“[将数据库服务器作为 Open Server 启动](#)”一节第 1060 页。

SQL Anywhere 服务器的缺省端口号为 2638。该端口号已由 Internet 编号授权委员会（Internet Assigned Numbers Authority，简称 IANA）分配给 SQL Anywhere，如果没有充分理由必须显式使用其它端口，建议使用此端口。

以下为有效的服务器地址条目：

```
elora,2638
123.85.234.029,2638
```

6. 单击 [OK]。

校验服务器地址

可以在 [Server Object] 菜单中使用 Ping 命令校验网络连接。

不校验数据库连接

通过验证网络连接，可以确认服务器是否在指定的计算机名和端口号接收请求。不校验有关数据库连接的任何情况。

◆ **对服务器执行 Ping 命令:**

1. 确保该数据库服务器正在运行。
2. 在 [DSEdit] 会话窗口的 [Server] 框中, 单击服务器条目。
3. 选择 [Server Object] » [Ping Server]。
4. 选择想要强制回应的地址并单击 [Ping]。

将出现一个窗口, 通知您连接是否成功。表示连接成功的窗口会显示打开连接和关闭连接都成功。

重命名服务器条目

可以重命名 [DSEdit] 会话窗口中的服务器条目。

◆ **重命名服务器条目:**

1. 在 [Server] 框中选择服务器条目。
2. 从 [Server Object] 菜单选择 [Rename]。
3. 在 [Server Name] 框中键入服务器条目的新名称。
4. 单击 [OK]。

删除服务器条目

可以在 [DSEdit] 会话窗口中删除服务器条目。

◆ **删除服务器条目:**

1. 在 [Server] 框中选择服务器条目。
2. 从 [Server Object] 菜单选择 [Delete]。

为 JDBC 配置服务器

JDBC 连接地址 (URL) 中包含了定位服务器所需的所有信息。请参见“[将 URL 提供给驱动程序](#)”一节 《SQL Anywhere 服务器 - 编程》。

Open Client 和 jConnect 连接的特性

当 SQL Anywhere 通过 TDS 为应用程序提供服务时，会自动将相关数据库选项的值设置为与 Adaptive Server Enterprise 的缺省行为兼容。这些选项是临时设置的，它们仅在连接期间有效。客户端应用程序可以随时替换它们。

缺省设置

在使用 TDS 的连接上设置的数据库选项包括：

选项	设置为
allow_nulls_by_default	Off
ansi_blanks	On
ansinull	Off
chained	Off
close_on_endtrans	Off
date_format	YYYY-MM-DD
date_order	MDY
escape_character	Off
isolation_level	1
on_tsq_error	Continue
quoted_identifier	Off
time_format	HH:NN:SS.SSS
timestamp_format	YYYY-MM-DD HH:NN:SS.SSS
tsql_variables	On

如何设置启动选项

TDS 连接的缺省数据库选项使用名为 `sp_tsql_environment` 的系统过程进行设置。此过程设置以下选项：

```
SET TEMPORARY OPTION allow_nulls_by_default='Off';
SET TEMPORARY OPTION ansi_blanks='On';
SET TEMPORARY OPTION ansinull='Off';
SET TEMPORARY OPTION chained='Off';
SET TEMPORARY OPTION close_on_endtrans='Off';
SET TEMPORARY OPTION date_format='YYYY-MM-DD';
```

```
SET TEMPORARY OPTION date_order='MDY';
SET TEMPORARY OPTION escape_character='Off';
SET TEMPORARY OPTION isolation_level='1';
SET TEMPORARY OPTION on_tsql_error='Continue';
SET TEMPORARY OPTION quoted_identifier='Off';
SET TEMPORARY OPTION time_format='HH:NN:SS.SSS';
SET TEMPORARY OPTION timestamp_format='YYYY-MM-DD HH:NN:SS.SSS';
SET TEMPORARY OPTION tsql_variables='On';
```

不要编辑 sp_tsql_environment 过程

不要自行改动 sp_tsql_environment 过程。该过程仅供系统使用。

该过程仅为使用 TDS 通信协议的连接设置选项。它包括 Open Client 和使用 jConnect 的 JDBC 连接。其它连接（ODBC 和嵌入式 SQL）使用数据库的缺省设置。

您可以更改 TDS 连接的选项。

◆ 更改 TDS 连接的选项设置：

1. 创建一个用于设置所需数据库选项的过程。例如，可以使用下面这样的过程：

```
CREATE PROCEDURE my_startup_procedure()
BEGIN
  IF CONNECTION_PROPERTY('CommProtocol')='TDS' THEN
    SET TEMPORARY OPTION quoted_identifier='Off';
  END IF
END;
```

此特殊过程示例将仅更改缺省设置中的 quoted_identifier 选项。

2. 将 login_procedure 选项设置为新过程的名称：

```
SET OPTION login_procedure= 'DBA.my_startup_procedure';
```

以后的连接将使用该过程。可以将该过程配置为因用户 ID 而异。

有关数据库选项的详细信息，请参见“数据库选项”一节第 466 页。

使用复制服务器复制数据

目录

将 SQL Anywhere 与复制服务器一起使用简介	1070
教程：使用复制服务器复制数据	1073
为复制服务器配置数据库	1081
使用 LTM	1084

将 SQL Anywhere 与复制服务器一起使用简介

复制服务器是一项基于连接的技术，用于实现事务的双向复制。它非常适合于在使用高速网络连接的少量企业数据库之间进行的复制，通常每个站点有一个管理员。在此类系统中，有可能将延迟时间缩短到几秒钟。

还可以使用 SQL Remote 复制 SQL Anywhere 数据，使用 MobiLink 同步数据。

请参见：

- “选择同步技术”一节 《SQL Anywhere 11 - 简介》
- “了解 MobiLink 同步” 《MobiLink - 入门》
- SQL Remote

开始之前

对打算将 SQL Anywhere 设置为参与复制服务器安装的复制服务器管理员，本章特别有用。您应该已了解复制服务器的文档，并熟悉复制服务器产品。本章并不介绍复制服务器本身。

有关复制服务器的信息（包括设计、命令和管理在内），请参见复制服务器的文档。

注意

SQL Anywhere 包含允许您在复制服务器系统中使用 SQL Anywhere 数据库的组件。复制服务器不包含在 SQL Anywhere 安装中。

需要单独授予许可的选项

任何作为主站点参与 Sybase 复制服务器安装的 SQL Anywhere 数据库，都需要日志传送管理器 (LTM)（它是 Sybase 复制服务器的 SQL Anywhere 复制代理）。LTM 所需的许可必须单独订购。如果 SQL Anywhere 用作复制站点，则不需要 LTM。

有关详细信息，请参见“单独授权的组件”一节 《SQL Anywhere 11 - 简介》。

复制服务器的特点

复制服务器为具有以下要求的复制系统而设计：

- **少量数据库** 复制服务器旨在支持服务器间的复制，其系统所包括的服务器数通常少于 100 个。
- **不间断连接** 主站点和复制站点可通过广域网进行连接，但复制服务器适用于以下情况：有一个近乎不间断的连接路径，用于在系统中的服务器间进行数据交换。
- **短等待时间** 短等待时间意味着，在一个数据库中输入数据后，再将数据复制到系统中每个数据库的延迟时间非常短。使用复制服务器，在主站点输入消息后，通常在几秒钟内便会发送复制消息。
- **高容量** 由于具有近乎不间断的连接和较高的性能，因此复制服务器适合于处理大容量的复制消息。

- **异类数据库** 复制服务器支持几种主流 DBMS，并且允许在复制过程中映射对象名，因而可支持异类数据库。

复制站点和主站点

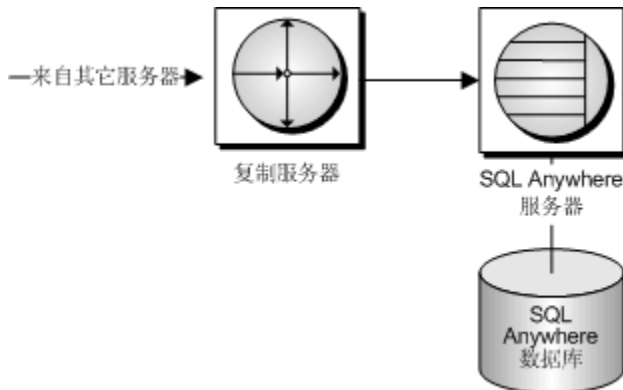
在安装复制服务器的过程中，将以**复制预订**的形式组织要在各个数据库之间共享的数据。

每个复制定义都有一个**主站点**，在复制中对数据进行的更改都发生在主站点中。在复制过程中接收数据的站点称为**复制站点**。

复制站点的组件

可以将 SQL Anywhere 用作复制站点。如果将 SQL Anywhere 用作复制站点，则不需要 LTM。

下图说明了 SQL Anywhere 作为复制站点参与复制服务器安装所需的组件。



- 复制服务器从主站点服务器接收数据更改。
- 复制服务器连接到 SQL Anywhere 以应用更改。
- SQL Anywhere 对数据库进行更改。

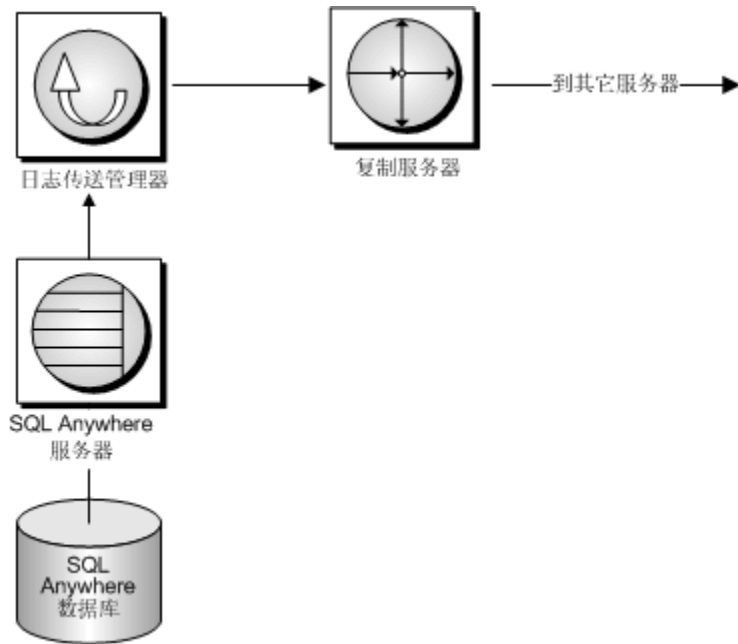
异步过程调用

复制服务器可以在复制站点上使用**异步过程调用**（Asynchronous Procedure Calls，简称 APC）来更改主站点数据库上的数据。如果您正在使用 APC，则以上图示不适用，而是与对主站点的要求相同。

主站点的组件

要将 SQL Anywhere 数据库用作主站点，需要使用日志传送管理器 (LTM)（SQL Anywhere 的复制代理）。LTM 支持复制服务器 10.0 及更高版本。请参见 <http://www.sybase.com/detail?id=1062617>。

下图说明了 SQL Anywhere 作为主站点参与复制服务器安装所需的组件。图中的箭头表示数据流。



- SQL Anywhere 数据库服务器用于管理数据库。
- SQL Anywhere 日志传送管理器连接到数据库。它将扫描事务日志以获取对数据所做的更改，然后将更改发送到复制服务器。
- 复制服务器将这些更改发送到复制站点数据库。

教程：使用复制服务器复制数据

本节提供了步骤详细的教程来介绍如何将数据从主数据库复制到复制数据库。本教程中的两个数据库都是 SQL Anywhere 数据库。

假定复制服务器已存在

本教程假定，您拥有一个正在 Windows 上运行的复制服务器，而且它与复制代理和 SQL Anywhere 全部都在同一台计算机上运行。

有关安装和配置复制服务器的详细信息，请参见复制服务器文档。

教程内容

本教程介绍了如何只对表进行复制。

有关复制过程的信息，请参见“[为复制准备过程和函数](#)”一节第 1085 页。

本教程使用一个（非常）原始的办公新闻系统作为示例：一个表，其中含有以下三个列：一个用于保存 ID 的整数列、一个用于保存新闻条目作者的用户 ID 的列、一个用于保存新闻条目的文本的列。id 列和 author 列构成主键。

在开始本教程之前，应先创建一个目录（例如 `c:\tutorial`）来保存您在教程中创建的文件。SQL Anywhere 日志扫描工具假定位于某个目录中的所有事务日志都属于同一个数据库，因此在本教程中，分别为每个数据库创建相应的目录（`c:\tutorial\primedb` 和 `c:\tutorial\repdb`），因为 LTM 会扫描脱机事务日志。

第 1 课：创建 SQL Anywhere 数据库

本节介绍了如何为复制创建并设置 SQL Anywhere 数据库。

可以使用 Sybase Central 或 dbinit 实用程序创建数据库。在本教程中，您使用 dbinit 实用程序。

◆ 创建主站点数据库：

- 在您创建的教程目录（例如 `c:\tutorial\primedb`）中运行以下命令，以保存主数据库：

```
dbinit primedb
```

这将在当前目录中创建名为 `primedb.db` 的数据库文件。

◆ 创建复制站点数据库：

- 在您创建的教程目录（例如 `c:\tutorial\repdb`）中运行以下命令，以保存复制数据库：

```
dbinit repdb
```

这将在当前目录中创建名为 `repdb.db` 的数据库文件。

下一步做什么？

接下来，您必须启动数据库服务器来运行这些数据库。

第 2 课：启动数据库服务器

您需要运行主站点数据库服务器，并装载主站点数据库。

◆ 启动主站点数据库服务器：

1. 转到教程目录。
2. 运行以下命令以启动运行 `primedb` 数据库的网络数据库服务器。您应该在缺省通信端口 (2638) 上使用 TCP/IP 网络通信协议：

```
dbsrv11 -x tcpip(port=2638) c:\tutorial\primedb\primedb.db
```

◆ 启动复制站点数据库服务器：

1. 转到教程目录。
2. 在另外一个端口上运行以下命令，以启动运行 `repdb` 数据库的网络数据库服务器：

```
dbsrv11 -x tcpip(PORT=2639) c:\tutorial\repdb\repdb.db
```

下一步做什么？

接下来，您必须在一个接口文件中为每个 SQL Anywhere 服务器创建条目，以便复制服务器能够与这些数据库服务器进行通信。

第 3 课：在系统中设置 Open Server

您需要向系统中的 Open Server 列表添加一组 Open Server。

添加 Open Server

Open Server 是使用 DSEdit 实用程序在接口文件 (*SQL.ini*) 中定义的。对于 Unix 用户，接口文件名为 *interfaces*，而实用程序名为 *sybinit*。

有关如何向接口文件添加定义的完整说明，请参见“配置 Open Server”一节第 1062 页。

所需的 Open Server

必须为每个 Open Server 定义提供一个**名称**和一个**地址**。不要变更每个定义的其他属性。您需要为以下的每一项添加一个 Open Server 条目：

● 主站点数据库 使用下列地址创建一个名为 PRIMEDB 的条目：

- 协议 NLWNSCK
- 网络地址 localhost,2638

● 复制站点数据库 使用如下地址创建一个名为 REPDB 的条目：

- 协议 NLWNSCK
- 网络地址 localhost,2639

- **主站点数据库上的 LTM** 这是必需的，这样您才可以正确地关闭 LTM。使用以下地址创建一个名为 PRIMELTM 的条目：
 - **协议** NLWNSCK
 - **网络地址** localhost,2640
- **您的复制服务器** 本教程假定您已定义了复制服务器的 Open Server。

下一步做什么？

接下来，请确认 Open Server 配置正确。

第 4 课：确认 Open Server 配置正确

可以通过从 DSEdit 实用程序中选择 [ServerObject] » [Ping Server] 来确认各个 Open Server 是否可用。

或者，您也可以通过使用 Open Client 应用程序（如 isql 实用程序）连接到数据库，以确认各个 Open Server 的配置是否正确。

要启动运行于主站点数据库中的 isql，请键入：

```
isql -U DBA -P sql -S PRIMEDB
```

注意

Open Client isql 实用程序与 SQL Anywhere Interactive SQL 实用程序并不相同。

第 5 课：将复制服务器信息添加到主数据库

要使数据库参与复制服务器安装，您需要将复制服务器表和过程添加到主站点数据库。您还需要创建两个供复制服务器使用的用户 ID。SQL 命令文件 *rssetup.sql* 随 SQL Anywhere 提供并执行这些任务。

必须通过 Interactive SQL 实用程序在 SQL Anywhere 服务器上运行 *rssetup.sql* 命令文件。

◆ 运行 rssetup 脚本：

1. 从 Interactive SQL 以具有 DBA 权限的用户身份连接到 SQL Anywhere 数据库。
2. 使用以下命令运行 rssetup 脚本：

```
read "install-dir\scripts\rssetup.sql"
```

此脚本中的 *install-dir* 为 SQL Anywhere 的安装目录。

或者，您也可选择 [文件] » [运行脚本]，然后浏览到该文件。

由 rssetup.sql 执行的操作

rssetup.sql 命令文件执行以下功能：

- 创建一个名为 **dbmaint** 的用户，口令为 **dbmaint**，该用户拥有 DBA 权限。这是复制服务器连接到主站点数据库所需的维护用户名和口令。
- 创建一个名为 **sa** 的用户，口令为 **sysadmin**，该用户拥有 DBA 权限。这是复制服务器在实现数据时所使用的用户 ID。
- 将 **sa** 和 **dbmaint** 添加到名为 **rs_systabgroup** 的组中。

口令和用户 ID

硬性规定的用户 ID (**dbmaint** 和 **sa**) 和口令可用于测试和在本教程中使用，但是在运行有安全性要求的数据库时，您需要更改口令，而且还可能需要更改用户 ID。拥有 DBA 权限的用户在 SQL Anywhere 数据库中拥有完全权限。

用户 ID **sa** 及其口令必须与复制服务器上系统管理员帐户的 ID 及其口令一致。SQL Anywhere 当前不接受 NULL 口令。

权限

rssetup.sql 脚本执行包括某些权限管理在内的若干操作。在此将概述由 *rssetup.sql* 做出的权限更改。您不需要自己进行这些更改。

为了能够进行复制，应确保 **dbmaint** 和 **sa** 用户无需显式指定表所有者便可访问需要复制的表。要做到这一点，表所有者用户 ID 必须拥有组成员权限，并且 **dbmaint** 和 **sa** 用户必须是表所有者组的成员。要授予组权限，您必须拥有 DBA 权限。

例如，如果用户 **DBA** 拥有该表，则您应向 **DBA** 授予组权限：

```
GRANT GROUP  
TO DBA;
```

然后，您应向 **dbmaint** 和 **sa** 用户授予 **DBA** 组的成员资格。要授予组成员资格，您必须拥有 **DBA** 权限或者必须是组 ID。

```
GRANT MEMBERSHIP  
IN GROUP "DBA"  
TO dbmaint ;  
GRANT MEMBERSHIP  
IN GROUP "DBA"  
TO sa;
```

第 6 课：为主数据库创建表

在本节中，您将在主站点数据库中使用 **isql** 创建一个表。首先，请确保您已连接到主站点数据库：

```
isql -U DBA -P sql -S PRIMEDB
```

然后在数据库中创建一个表：

```
CREATE TABLE news (  
  ID INT,  
  AUTHOR CHAR( 128 ) DEFAULT CURRENT USER,  
  TEXT CHAR( 255 ),  
  PRIMARY KEY ( ID, AUTHOR )  
)  
go
```

标识符的大小写敏感性

在 SQL Anywhere 中，所有标识符均不区分大小写。在 Adaptive Server Enterprise 中，标识符按缺省设置是区分大小写的。为了能与 Adaptive Server Enterprise 兼容，请确保在 SQL Anywhere 中您的标识符与 SQL 语句中的所有部分匹配。

在 SQL Anywhere 中，口令始终区分大小写。作为标识符，用户 ID 在所有 SQL Anywhere 数据库中均不区分大小写。

有关详细信息，请参见“[CREATE DATABASE 语句](#)”一节《[SQL Anywhere 服务器 - SQL 参考](#)》。

为了使 news 成为复制主站点的一部分，您必须对该表使用 ALTER TABLE 语句，将 REPLICATE 设置为 ON：

```
ALTER TABLE news
REPLICATE ON
go
```

这相当于在 Adaptive Server Enterprise 的表上运行 sp_setreplicate 或 sp_setreptable 过程。不能在 CREATE TABLE 语句中设置 REPLICATE ON。

第 7 课：将复制服务器信息添加到复制数据库

您应该在复制数据库上采用与在主数据库上完全相同的方式运行 *rssetup.sql* 命令文件。

这些任务与在主数据库上执行的任务相同。

有关完整说明，请参见“[第 5 课：将复制服务器信息添加到主数据库](#)”一节第 1075 页。

第 8 课：为复制数据库创建表

复制站点数据库需要用表来保存它所接收的数据。您可以现在就创建这些表。创建好所需的数据库元素后，不需要额外语句将它们变为复制服务器系统的复制站点。尤其是，您不需要将 REPLICATE 设置为 ON，只有在主站点上才必须这样做。

复制服务器支持在具有不同名称的表和列之间进行复制。但是，作为一个简单的示例，可在复制数据库中创建一个表，该表在定义上与主数据库中的表完全相同（REPLICATE 除外，它在复制数据库中不设置为 ON）。创建该表的语句为：

```
CREATE TABLE news (
  ID INT,
  AUTHOR CHAR( 40 ) DEFAULT CURRENT USER,
  TEXT CHAR( 255 ),
  PRIMARY KEY ( ID, AUTHOR )
)
go
```

在本教程中，CREATE TABLE 语句必须与主站点的语句完全相同。

必须确保用户 dbmaint 和 sa 无需指定所有者名称便可访问此表。另外，这些用户 ID 必须在该表上拥有 SELECT 和 UPDATE 权限。

第 9 课：设置复制服务器

您需要在复制服务器上执行以下任务：

- 为主站点数据库服务器创建一个连接。
- 为复制站点数据库服务器创建一个连接。
- 创建一个复制定义。
- 创建对该复制的预订。
- 启动 SQL Anywhere LTM。

为主站点创建连接

使用 `isql` 连接到复制服务器，并创建与主站点 SQL Anywhere 数据库的连接。

以下命令用于创建与 PRIMEDB Open Server 上 `primedb` 数据库的连接。

```
CREATE CONNECTION TO PRIMEDB.primedb
SET ERROR CLASS rs_sqlserver_error_class
SET FUNCTION STRING class rs_sqlserver_function_class
SET USERNAME dbmaint
SET PASSWORD dbmaint
WITH LOG TRANSFER ON
go
```

如果您已在 `rssetup.sql` 命令文件中更改了用户 ID `dbmaint` 和口令，请确保在此命令中替换 `dbmaint` 用户名和口令。

实际上，复制服务器并不使用 `primedb` 数据库名；使用的数据库名是从 PRIMEDB Open Server 的命令行读取的。不过，您必须在 `CREATE CONNECTION` 语句中添加一个数据库名才能符合语法。

有关创建连接语句的完整说明，请参见 *Replication Server Reference Manual* 中的 "Replication Server Commands" 一章。

为复制站点创建连接

使用 `isql` 连接到复制服务器，并创建与复制站点 SQL Anywhere 数据库的连接。

以下命令用于创建与 REPDB Open Server 上 `repdb` 数据库的连接。

```
CREATE CONNECTION TO REPDB.repdb
SET ERROR CLASS rs_sqlserver_error_class
SET FUNCTION STRING CLASS rs_sqlserver_function_class
SET USERNAME dbmaint
SET PASSWORD dbmaint
go
```

该语句与主站点服务器语句的不同之处在于：在该语句中没有 `WITH LOG TRANSFER ON` 子句。

如果您已在 `rssetup.sql` 命令文件中更改了用户 ID `dbmaint` 和口令，请确保在此命令中替换 `dbmaint` 用户名和口令。

创建复制定义

使用 `isql` 连接到复制服务器，并创建复制定义。以下语句为 `primedb` 数据库中的 `news` 表创建复制定义：

```
CREATE REPLICATION DEFINITION news
WITH PRIMARY AT PRIMEDB.primedb
( id INT, author CHAR( 128 ), text CHAR(255) )
PRIMARY KEY ( id, author )
go
```

有关 `CREATE REPLICATION DEFINITION` 语句的完整说明，请参见 *Replication Server Reference Manual*。

如果您在 LTM 配置文件中将 `qualify_table_owners` 选项设置为 ON，则必须在该语句中为所有正在复制的表指定表所有者。

配置和启动 SQL Anywhere LTM

要进行复制，必须对主站点服务器运行 SQL Anywhere LTM。在启动 SQL Anywhere LTM 之前，请通过编辑 LTM 配置文件来确保它已被正确配置。

以下是 `primedb` 数据库的一个示例配置文件。如果您正在按本教程中的步骤执行，请制作一个名为 `primelstm.cfg` 的该文件的副本：

```
#
# Configuration file for 'PRIMELSTM'
#
SQL_server=PRIMEDB
SQL_database=primedb
SQL_user=sa
SQL_pw=sysadmin
RS_source_ds=PRIMEDB
RS_source_db=primedb
RS=your-rep-server-name-here
RS_user=sa
RS_pw=sysadmin
LTM_admin_user=DBA
LTM_admin_pw=sql
LTM_charset=cp850
scan_retry=2
APC_user=sa
APC_pw=sysadmin
SQL_log_files=C:\TUTORIAL\PRIMEDB
```

如果已经在 `rssetup.sql` 命令文件中将 `sa` 和 `sysadmin` 更改为其它用户 ID 和口令，则您应在此配置中使用新的用户 ID 和口令。

要启动在主站点服务器上运行的 SQL Anywhere LTM，请输入以下命令：

```
dblstm -S PRIMELSTM -C primelstm.cfg
```

连接信息位于 `primelstm.cfg` 中。在此命令中，PRIMELSTM 为 LTM 的服务器名。

通过键入以下语句，您可以找到有关 SQL Anywhere LTM 的使用信息：

```
dblstm -?
```

可以将 SQL Anywhere LTM 作为 Windows 服务来运行。

有关将程序作为服务运行的信息，请参见“[在当前会话外部运行服务器](#)”一节第 57 页。

为复制创建预订

使用 isql 连接到复制服务器，并为复制创建预订。

以下语句为 news 复制创建了一个预订，并将复制站点作为 repdb 数据库。

有关 news 复制的信息，请参见“[创建复制定义](#)”一节第 1079 页。

```
CREATE SUBSCRIPTION NEWS_SUBSCRIPTION
FOR news
WITH REPLICATE AT REPDB.repdb
go
```

现在您已完成安装。尝试复制数据以确认所安装的系统工作正常。

第 10 课：在主站点输入用于复制的数据

您现在可以将数据从主数据库复制到复制数据库。例如，使用 isql 实用程序连接到主数据库，并在 news 表中输入一行。

```
INSERT news (id, text)
VALUES (1, 'Test news item.')
COMMIT
go
```

SQL Anywhere LTM 只将已提交的更改发送到复制服务器。对数据的更改会在 LTM 下次轮询事务日志时进行复制。

检查已发送到 repdb 的数据，方法是使用 isql 实用程序连接到复制数据库，然后执行以下 SQL 语句：

```
SELECT * FROM news
go
```

教程结束

您现在已经完成了本教程。

为复制服务器配置数据库

每个 SQL Anywhere 数据库在可以参与复制服务器安装之前都需要进行配置。配置数据库的过程涉及以下任务：

- 为维护用户选择一个安全的用户 ID 并选择一个在复制服务器实现数据时用的用户名。
- 为复制服务器设置数据库。
- 在需要的地方配置语言和字符集。

配置 LTM

每个主站点 SQL Anywhere 数据库都需要有 LTM 来向复制服务器发送数据。每个主站点或复制站点 SQL Anywhere 数据库都需要一个 Open Server 定义，以便复制服务器能够连接到数据库。

有关配置 LTM 的信息，请参见“配置 LTM”一节第 1086 页。

为复制服务器设置数据库

创建了 SQL Anywhere 数据库，并创建了该数据库中必需的表等对象后，即可配置该数据库以供复制服务器使用。您可以使用随 SQL Anywhere 复制代理产品提供的安装脚本来实现这一点。该脚本名为 *rssetup.sql*。

何时需要运行安装脚本

需要在任何参与复制服务器安装的 SQL Anywhere 数据库上运行安装脚本，无论它是作为主站点还是作为复制站点。

安装脚本的作用是什么

安装脚本用于创建复制服务器在连接到数据库时所需的用户 ID。它还创建一组由复制服务器使用的存储过程和表。这些表的名称以字符 **rs_** 开头，而过程名以字符 **sp_** 开头。其中某些过程对字符集和语言配置非常重要。

准备运行安装脚本

对于包含被复制的表的每个本地数据库，复制服务器使用一个特殊的数据服务器**维护用户**登录名。这样，复制服务器便可维护和更新数据库中被复制的表。

维护用户

安装脚本会创建一个维护用户，其用户名为 **dbmaint**，口令为 **dbmaint**。维护用户在 SQL Anywhere 数据库中拥有 DBA 权限，这样该用户就可以完全控制数据库。出于安全考虑，您应该更改维护用户 ID 和口令。

◆ 更改维护用户 ID 和口令:

1. 在文本编辑器中打开 *rssetup.sql* 安装脚本。该脚本保存在 SQL Anywhere 安装目录的 *scripts* 子目录中。
2. 将出现的所有 dbmaint 用户 ID 都更改为您选择的新维护用户 ID。
3. 将口令 dbmaint 更改为您所选择的新维护用户口令。口令位于安装脚本文件顶部的以下位置:

```
GRANT CONNECT TO dbmaint  
IDENTIFIED BY dbmaint;
```

实现用户 ID

当复制服务器连接到数据库上以实现参与复制的数据的初始副本时，它是通过使用复制服务器系统管理员帐户来完成此项工作的。

SQL Anywhere 数据库的用户 ID 和口令必须与复制服务器系统管理员的用户 ID 和口令相符。SQL Anywhere 不接受 NULL 口令。

安装脚本假定复制服务器管理员的用户 ID 为 sa，口令为 **sysadmin**。您应该更改此用户 ID 和口令以与实际的用户名和口令匹配。

◆ 更改系统管理员用户 ID 和口令:

1. 在文本编辑器中打开 *rssetup.sql* 安装脚本。
2. 更改所出现的所有 sa 用户 ID，以与复制服务器系统管理员用户 ID 相匹配。
3. 更改 sa 用户的口令，以与复制服务器系统管理员口令相匹配。
口令的初始设置为 **sysadmin**。

运行安装脚本

相应地修改安装脚本以使用户 ID 和口令匹配后，即可运行安装脚本，以便在 SQL Anywhere 数据库中创建维护用户和系统管理员用户。

◆ 运行安装脚本

1. 在 SQL Anywhere 数据库服务器上启动 SQL Anywhere 数据库。
2. 启动 Interactive SQL 实用程序，然后以具有 DBA 权限的用户身份连接到数据库。
创建 SQL Anywhere 数据库时，它包含用户 ID **DBA** 和口令 **sql**。该用户拥有 DBA 权限。
3. 通过在 [SQL 语句] 窗格中输入下列命令来运行脚本:

```
read install-dir\scripts\rssetup.sql
```

此命令中的 *install-dir* 为 SQL Anywhere 的安装目录。

复制服务器字符集和语言问题

每个 SQL Anywhere 数据库在创建时都被指派了特定的归类（字符集和排序顺序）。复制服务器对字符集和排序顺序使用另外一组标识符。

在 LTM 配置文件中设置字符集和语言参数。如果不能确定要指定的字符集标签，可执行以下操作来确定服务器的字符集：

◆ 确定字符集：

- 执行下列命令：

```
exec sp_serverinfo csname
```

有关语言标签的列表，请参见“语言标签值”一节第 389 页。

复制服务器中的标识符

如果您将 SQL Anywhere 复制代理与复制服务器 15.0 和 Open Client/Open Server 15.0 配合使用，则该复制代理可支持长达 128 个字节的表名、列名、过程名、函数名和参数名。

在将该复制代理与旧版本的复制服务器和 Open Client/Open Server 配合使用时，标识符的最大长度为 30 字节。

使用 LTM

由于 SQL Anywhere LTM 依赖于 SQL Anywhere 事务日志中的信息，所以在没有存储备份的情况下（例如，使用事务日志镜像），请注意不要删除或损坏事务日志。

有关事务日志管理的详细信息，请参见“事务日志和备份管理”一节第 1090 页。

因为事务日志的格式不同，所以不能以 Adaptive Server Enterprise LTM 替换 SQL Anywhere LTM。

SQL Anywhere LTM 支持对插入、更新和删除操作的复制，以及对使用 Transact-SQL 方言的存储过程的调用的复制。

Adaptive Server Enterprise LTM 在提交数据更改之前，会先将其发送到复制服务器。复制服务器会在 COMMIT 语句到达之前一直保存更改。相反，SQL Anywhere LTM 只将已提交的更改发送到复制服务器。对于时间较长的事务，这或许会导致复制过程中的延迟增大，因为所有的更改在分发之前都必须通过复制服务器。

为复制配置表

可使用 `sp_setreplicate` 或 `sp_setrepproc` 系统过程或者 ALTER TABLE 语句配置用于复制的表。通过使用带有一个子句的 ALTER TABLE 语句将表标识为主数据源：

```
ALTER TABLE table-name
SET REPLICATE ON;
```

为表设置 REPLICATE ON 的效果

设置 REPLICATE ON 会在事务日志中放入额外的信息。每当表上发生 UPDATE、INSERT 或 DELETE 操作时，SQL Anywhere 复制代理使用此额外信息，在需要之处将行的完整前映像提交给复制服务器以进行复制。

即使只需要复制表中的某些数据，对表所做的所有更改也都会被提交到复制服务器。复制服务器负责将要复制的数据与不需要复制的数据区分开。

当您更新、插入或删除行时，行的前映像就是操作之前该行的内容，而后映像则是操作之后该行的内容。对于 INSERTS，只提交后映像（其前映像为空）。对于 DELETES，后映像为空，且只提交前映像。对于 UPDATES，前映像和更新值都会被提交。

复制支持下列数据类型：

数据类型	描述（Open Client/Open Server 类型）
精确的整数数据类型	int、smallint 和 tinyint
精确的小数数据类型	decimal 和 numeric
近似的数字数据类型	float（8 字节）和 real
货币数据类型	money 和 smallmoney

数据类型	描述（Open Client/Open Server 类型）
字符数据类型	char(n)、varchar(n) 和 text
日期和时间数据类型	datetime 和 smalldatetime
二进制数据类型	binary(n)、varbinary(n) 和 image
位数据类型	bit

注意

SQL Anywhere 支持零长度的非 NULL 数据。但是，零长度的非空 LONG VARCHAR 和 LONG BINARY 数据在复制到复制站点时会成为 NULL。

当主表中某些列的数据类型不受支持时，如果您使用兼容的受支持数据类型创建复制定义，则可复制数据。例如，要复制 DOUBLE 列，可在复制定义中将该列定义为 FLOAT。

为表设置 REPLICATE ON 的副作用

对于进行了大量更新的表，复制性能会受到影响。如果您遇到可能与复制通信流量相关的性能问题，则可考虑使用复制的过程，因为复制的过程仅发送复制过程调用，而不发送过程中的每一项操作。

因为设置 REPLICATE ON 会向事务日志发送额外信息，所以日志的增长要快于非复制数据库日志的增长。

最少列复制定义

SQL Anywhere LTM 支持复制服务器的复制最少列功能。该功能在复制服务器中已启用。

有关复制最少列的详细信息，请参见复制服务器的文档。

为复制准备过程和函数

可使用存储过程来修改表中的数据。更新、插入和删除操作将从过程内部执行。

只要过程满足特定的条件，复制服务器便可复制它们。过程中的第一条语句必须对将要被复制的过程执行更新。

有关复制服务器如何进行过程复制的完整说明，请参见复制服务器文档。

SQL Anywhere 支持用于存储过程的以下两种方言：基于 ISO/ANSI 标准草案的 Watcom-SQL 方言及 Transact-SQL 方言。必须使用 Transact-SQL 来编写将要复制的存储过程。

函数 APC 格式

SQL Anywhere LTM 支持复制服务器的函数 APC 格式。要使用这些函数，需将配置参数 `rep_func` 设置为 `on`（而缺省值为 `off`）。

LTM 将所有复制的 APC 解释为表 APC 或函数 APC。单个 SQL Anywhere 数据库不可以将函数 APC 与其它表 APC 组合起来。

有关复制函数的详细信息，请参见复制服务器的文档。

用于控制过程复制的 SQL 语句

使用 ALTER PROCEDURE 语句可将过程配置成为复制源。

以下语句使过程 MyProc 充当复制源。

```
ALTER PROCEDURE MyProc  
REPLICATE ON;
```

以下语句禁止过程 MyProc 用作复制源。

```
ALTER PROCEDURE MyProc  
REPLICATE OFF;
```

也可使用 sp_setreplproc 或 sp_setrepproc 系统过程设置用于复制的过程。

为过程设置 REPLICATE ON 的效果

在过程被用作复制数据源时，调用该过程会向事务日志发送额外信息。

异步过程

异步过程是为更新主站点数据库的数据而在复制站点数据库中调用的过程。异步过程不在复制站点执行任何操作，而是将对过程的调用复制到主站点上，在主站点上执行同名的过程。这种现象被称为**异步过程调用**（Asynchronous Procedure Calls，简称 APC）。由 APC 所做的更改随后以普通方式从主数据库复制到复制数据库。

有关 APC 的信息，请参见复制服务器的文档。

APC_user 和 APC 支持

在 SQL Anywhere 与 Adaptive Server Enterprise 这两者中，对 APC 的支持是不同的。在 Adaptive Server Enterprise 中，执行各个 APC 所使用的用户 ID 和口令都是在复制站点上对过程进行调用的用户的用户 ID 和口令。而在 SQL Anywhere 中，事务日志并不存储口令，因此口令在主站点上不可用。为了解决这种差异，LTM 配置文件会保存一个用户 ID 及相关口令，主站点上过程的执行将使用此用户 ID (**APC_user**)。因此，APC_user 必须在主站点上针对每个可能会被调用的 APC 拥有适当权限。

配置 LTM

您可通过修改 LTM **配置文件**来控制 LTM 的行为，该配置文件是使用文本编辑器创建和编辑的纯文本文件。LTM 配置文件包含了 LTM 所需的信息，例如，它要从中传送日志的 SQL Anywhere 服务器，以及它要将日志传送到的复制服务器。要运行 LTM，您需要一个有效的配置文件。

创建配置文件

必须先使用文本编辑器创建一个配置文件，然后才能运行 LTM。-C LTM 命令指定要使用的配置文件的名称，缺省名称为 *dbltm.cfg*。

配置文件的格式

LTM 配置文件与 *Replication Server Administration Guide* 中描述的复制服务器配置文件的格式相同。可归纳为：

- 配置文件的每行都包含一个条目。
- 条目包括一个参数，后面跟 = 字符，再后面跟值：

```
Entry=value
```

- 以 # 字符开头的行是注释，LTM 将忽略注释。
- 配置文件不能包含前导空白。
- 条目是区分大小写的。

有关可用配置文件参数的完整列表，请参见“[LTM 配置文件](#)”一节第 749 页。

示例配置文件

- 以下是一个示例 SQL Anywhere LTM 配置文件。

```
# This is a comment line
# Names are case sensitive.
SQL_user=sa
SQL_pw=sysadmin
SQL_server=PRIMESV
SQL_database=primedb
RS_source_ds=PRIMESV
RS_source_db=primedb
RS=MY_REPSEVER
RS_user=sa
RS_pw=sysadmin
LTM_admin_user=DBA
LTM_admin_pw=sql
LTM_charset=cp850
scan_retry=2
SQL_log_files=e:\logs\old_logs
APC_user=sa
APC_pw=sysadmin
```

批量复制事务

缓冲事务的效果

LTM 允许将复制命令缓冲到复制服务器。缓冲复制命令并将其成批发送，不仅会使得发送的消息量减少，而且还会使总吞吐量大大提高，特别是对大容量处理的安装。

批处理模式的工作原理

缺省情况下，LTM 会缓冲事务。当缓冲区出现以下情况时，缓冲区会刷新（即，事务发送给复制服务器）：

- **达到命令的最大数** `batch_ltl_sz` 参数可设置刷新前存储在缓冲区中的 LTL（日志传送语言）命令的最大数。缺省设置为 200。
- **达到最大可使用内存** `batch_ltl_mem` 参数可设置缓冲区在刷新前可以占用的最大内存。缺省设置为 256 KB。
- **完成事务日志处理** 如果事务日志中没有其它条目需要处理（即，LTM 已处理完所有已提交的事务），则缓冲区就会刷新。

关闭缓冲

可以通过将 `batch_ltl_cmds` 参数设置为 `off` 来关闭对事务的缓冲：

```
batch_ltl_cmds=off
```

语言和字符集问题

语言和字符集问题在许多复制站点中是很重要的考虑事项。系统中的每个数据库和服务器都使用一个特定的归类（字符集和排序顺序）对字符串进行存储和排序。SQL Anywhere 字符集支持的行为方式，与 Adaptive Server Enterprise 及其它基于 Open Client/Open Server 的应用程序的字符集支持的行为方式不同。

本节将介绍如何配置 SQL Anywhere LTM，以使 SQL Anywhere 数据库中的数据可与复制服务器以及其它数据库共享。

LTM 自动使用缺省的 Open Client/Open Server 语言、排序顺序和字符集。您可以通过将条目添加到 LTM 配置文件来覆盖这些缺省值。

Open Client/Open Server 归类

Adaptive Server Enterprise、复制服务器和其它 Open Client/Open Server 应用程序对字符集进行管理的方法相同。

有关 Open Client/Open Server 字符集支持的信息，请参见 Adaptive Server Enterprise *System Administration Guide* 中的 "Configuring Character Sets, Sort Orders, and Languages" 一章。

有关复制服务器中字符集问题的详细信息，请参见 *Replication Server Design Guide* 中的 "International Replication Design Considerations" 一章。

本节简要概述 Open Client/Open Server 对字符集的支持。

国际化文件

以特定语言支持数据处理的文件称为 **国际化文件**。Adaptive Server Enterprise 及其它 Open Client/Open Server 应用程序都附带了多种国际化文件。

在 Sybase 目录下包含一个名为 *charsets* 的目录。*Charsets* 具有一组子目录，分别对应您可以使用的各个字符集。每个字符集都包含一组文件，如下表所述

文件	说明
<i>Charset.loc</i>	字符集定义文件，这些文件定义了各个字符的词法属性，例如数字和字母、标点符号、操作数、大写或小写。
*.srt	为数字和字母以及特殊字符定义排序顺序。
*.xlt	与实用程序一起使用的终端特有的字符转换文件。

LTM 配置文件中的字符集设置

LTM 配置文件中有三项设置涉及字符集问题：

- **LTM_charset** 供 LTM 使用的字符集。您可以指定 Sybase 支持的任何字符集。
- **LTM_language** LTM 用来向错误日志及其客户端打印消息的**语言**。可以选择本地化 LTM 支持的任何语言，只要该语言与 LTM 字符集兼容即可。
SQL Anywhere LTM 已被本地化为多种语言。
- **LTM_sortorder** LTM 用于比较用户名的 Open Client/Open Server 排序顺序。您可以指定与 LTM 的字符集兼容的任何受 Adaptive Server Enterprise 支持的排序顺序。复制系统中的所有排序顺序都应该相同。缺省排序顺序是二进制排序。

注意

字符集 在 Open Client/Open Server 环境中，LTM 应该与数据库服务器和附加到该服务器的复制服务器使用同一个字符集。

指定的 SQL Anywhere 字符集不同于 Open Client/Open Server 字符集，因此要求 SQL Anywhere 字符集必须与 LTM 字符集兼容。

语言 Sybase 发布目录的 *locales* 子目录中的 *locales.dat* 文件包含有效的映射设置。不过，在用户界面内的 LTM 输出消息当前使用支持的本地化语言。

排序顺序 复制系统中的所有排序顺序都应该相同。可在 Sybase 发布目录 *locales* 子目录中的 *locales.dat* 文件内找到适用于您的平台的缺省条目。

示例

- 下面的设置对于日文系统有效：

```
LTM_charset=SJIS
LTM_language=Japanese
```

事务日志和备份管理

Adaptive Server Enterprise LTM 和 SQL Anywhere LTM 之间的区别之一是：Adaptive Server Enterprise LTM 依赖于临时恢复数据库来访问旧事务，而 SQL Anywhere LTM 则依赖于对旧事务日志的访问。对 SQL Anywhere LTM 而言，并不存在临时恢复数据库。

复制依赖于对事务日志中操作的访问，但对于 SQL Anywhere 主站点数据库，复制有时依赖于对旧事务日志的访问。本节介绍如何在 SQL Anywhere 主站点设置备份过程，以确保对旧事务日志的正确访问。

丢失事务日志的后果

在 SQL Anywhere 主数据库站点上经常进行备份至关重要。丢失事务日志可能意味着需要重新实现复制站点数据库。建议在主数据库站点上使用事务日志镜像。

有关事务日志镜像以及其它备份过程的信息，请参见“事务日志镜像”一节第 15 页和“备份和数据恢复”第 817 页。

LTM 配置文件包含一个目录条目，该条目指向保存备份事务日志的目录。本节将介绍如何设置备份过程以确保此类目录保持正确状态。

备份实用程序选项

在备份实用程序中，您可以选择在备份和重新启动时重命名事务日志。对于 dbbackup 实用程序，该选项是 -r 选项。建议在备份主数据库事务日志和复制数据库事务日志时使用此选项。

例如，请考虑以下情形：在目录 *c:\prime* 中有一个名为 *primedb.db* 的数据库，并且在目录 *d:\primelog\primedb.log* 中有一个事务日志。使用重命名和重新启动选项将此事务日志备份到目录 *e:\primebak* 的过程需要执行以下任务：

1. 创建备份文件 *e:\primebak\primedb.log*，备份事务日志。
2. 将现有事务日志重命名为 *d:\primelog\YYMMDDxx.log*，其中 **xx** 为从 **AA** 到 **ZZ** 的连续字符。
3. 启动一个新事务日志，如 *d:\primelog\primedb.log*。

几次备份之后，目录 *d:\primelog* 将包含一组有序的事务日志。日志目录不应包含由此备份过程生成的一系列日志之外的任何事务日志。

4. 修改 LTM 配置文件并将 `SQL_log_files` 设置为 *d:\primelog*，而非 *e:\primebak*。*e:\primebak* 目录仅用于恢复，不用于日志扫描。

使用 delete_old_logs 选项

缺省情况下，SQL Anywhere 数据库的 `delete_old_logs` 选项被设置为 Off。如果将缺省值设置为 on，则 LTM 会在复制服务器不再需要访问事务时自动删除旧事务日志。此选项有助于在复制设置中管理磁盘空间。

例如，为 PUBLIC 组设置 `delete_old_logs` 选项：

```
SET OPTION PUBLIC.delete_old_logs = 'ON';
```

或者

```
SET OPTION PUBLIC.delete_old_logs = '10 days';
```

有关详细信息，请参见“[delete_old_logs 选项 \[MobiLink 客户端\] \[SQL Remote\] \[复制代理\]](#)”一节第 502 页。

卸载实用程序和复制

如果数据库参与复制，则在卸载和重装时必须谨慎，以避免重新实现数据库。复制是基于事务日志进行的，而卸载和重装数据库会删除旧事务日志。有关重建参与复制的数据库的信息，请参见“[重建参与同步或复制的数据库](#)”一节《[SQL Anywhere 服务器 - SQL 的用法](#)》。

复制整个数据库

SQL Anywhere 为复制整个数据库提供了一个捷径，因此您不必将数据库中的每个表都设置为被复制的表。

可以使用 SET OPTION 语句设置名为 replicate_all 的 PUBLIC 数据库选项。可以使用以下命令指定对整个数据库进行复制：

```
SET OPTION PUBLIC.replicate_all='On';
```

您需要具有 DBA 权限才能更改此设置及其它 PUBLIC 选项设置。必须重新启动数据库才能使新设置生效。replicate_all 选项对过程无效。请参见“[replicate_all 选项 \[复制代理\]](#)”一节第 539 页。

停止 LTM

在 Windows 中，可通过用户界面关闭 LTM；在其它环境中，可通过发出命令来关闭 LTM。

◆ 当 LTM 未作为服务运行时，停止 Windows 上的 LTM

- 在用户界面上单击 [关闭]。

◆ 通过发出命令停止 LTM

1. 使用 LTM 配置文件中的 LTM_admin_user 登录名和口令从 isql 连接到 LTM。用户 ID 和口令是区分大小写的。
2. 使用 SHUTDOWN 语句停止 LTM。

示例

以下语句将 isql 连接到 LTM PRIMELTM 并将其关闭：

```
isql -SPRIMELTM -UDBA -Psql
1> shutdown
2> go
```

术语表

术语表	1095
-----------	------

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 1112 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 1122 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 1122 页](#)
- [“代码页”一节第 1097 页](#)
- [“归类”一节第 1101 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 1109 页](#)
- [“隔离级别”一节第 1100 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 1122 页
- “外键”一节第 1114 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 1113 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 1112 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 1099 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 1101 页](#)
- [“语句级触发器”一节第 1119 页](#)
- [“完整性”一节第 1115 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 1120 页](#)

代理 ID

另请参见：[“客户端消息存储库 ID”一节第 1105 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 1122 页](#)
- [“编码”一节第 1095 页](#)
- [“归类”一节第 1101 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 1111 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 1112 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 1113 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 1099 页](#)
- [“项目” 一节第 1117 页](#)
- [“发布更新” 一节第 1098 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 1099 页
- “发布”一节第 1098 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 1099 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 1099 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 1106 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 1106 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 1099 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 1099 页](#)
- [“发布更新”一节第 1098 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 1105 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 1111 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 1122 页
- “代码页”一节第 1097 页
- “编码”一节第 1095 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 1097 页
- “语句级触发器”一节第 1119 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 1109 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 1102 页
- “jConnect”一节第 1102 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 1102 页](#)
- [“iAnywhere JDBC 驱动程序”一节第 1101 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 1102 页](#)
- [“iAnywhere JDBC 驱动程序”一节第 1101 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 1105 页](#)
- [“视图”一节第 1109 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 1099 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 1121 页](#)
- [“外键约束”一节第 1115 页](#)
- [“主键约束”一节第 1122 页](#)
- [“唯一约束”一节第 1116 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 1109 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 1110 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 1114 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 1105 页](#)
- [“全局临时表”一节第 1108 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 1100 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 1105 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 1099 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 1105 页](#)
- [“生成的连接条件”一节第 1110 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 1104 页](#)
- [“全局临时表”一节第 1108 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 1100 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 1099 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 1113 页](#)
- [“同步”一节第 1113 页](#)
- [“UltraLite”一节第 1114 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 1105 页](#)
- [“外连接”一节第 1115 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 1099 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 1105 页](#)
- [“局部临时表”一节第 1104 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 1109 页](#)
- [“事务日志镜像”一节第 1110 页](#)
- [“完全备份”一节第 1115 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 1099 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 1102 页](#)
- [“视图”一节第 1109 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 1103 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 1109 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 1105 页](#)
- [“连接条件”一节第 1105 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 1119 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 1114 页](#)
- [“主键”一节第 1122 页](#)
- [“数据库管理系统 \(DBMS\)”一节第 1111 页](#)
- [“关系数据库管理系统 \(RDBMS\)”一节第 1100 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 1100 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 1112 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 1111 页
- “SYS” 一节第 1113 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 1098 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 1107 页
- “SQL” 一节第 1112 页
- “数据库管理系统 (DBMS)” 一节第 1111 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 1100 页](#)
- [“并发”一节第 1095 页](#)
- [“完整性”一节第 1115 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 1113 页](#)
- [“复制”一节第 1099 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 1099 页](#)
- [“监听器”一节第 1103 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 1106 页](#)
- [“SQL Remote”一节第 1112 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 1099 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 1108 页](#)
- [“服务器启动的同步”一节第 1099 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 1114 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 1122 页](#)
- [“外表”一节第 1114 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 1121 页](#)
- [“检查约束”一节第 1103 页](#)
- [“主键约束”一节第 1122 页](#)
- [“唯一约束”一节第 1116 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 1105 页](#)
- [“内连接”一节第 1107 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 1121 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 1096 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 1099 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 1100 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 1115 页](#)
- [“主键约束”一节第 1122 页](#)
- [“约束”一节第 1121 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 1116 页](#)。

Windows Mobile

Microsoft 为移动设备制造的操作系统的系列。

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 1103 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 1099 页](#)
- [“发布”一节第 1098 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 1104 页](#)
- [“服务器消息存储库”一节第 1099 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 1099 页](#)
- [“统一数据库”一节第 1113 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 1099 页](#)
- [“FILE”一节第 1099 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 1121 页](#)
- [“用户定义数据类型”一节第 1119 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 1114 页](#)

用户定义数据类型

请参见“域”一节第 1119 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 1119 页
- “游标位置”一节第 1119 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 1119 页
- “游标位置”一节第 1119 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 1119 页
- “游标结果集”一节第 1119 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 1097 页
- “行级触发器”一节第 1101 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 1110 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 1098 页](#)
- [“MobiLink 用户”一节第 1107 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 1107 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 1098 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 1113 页](#)
- [“统一数据库”一节第 1113 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 1103 页](#)
- [“外键约束”一节第 1115 页](#)
- [“主键约束”一节第 1122 页](#)
- [“唯一约束”一节第 1116 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 1099 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 1109 页](#)

争用

为获取资源而竞争的行为。例如，就数据库而言，如果有两个或更多个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 1113 页](#)
- [“基于 SQL 的同步”一节第 1103 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 1114 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 1121 页](#)
- [“检查约束”一节第 1103 页](#)
- [“外键约束”一节第 1115 页](#)
- [“唯一约束”一节第 1116 页](#)
- [“完整性”一节第 1115 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 1097 页](#)
- [“编码”一节第 1095 页](#)
- [“归类”一节第 1101 页](#)

索引

其它

-? 服务器选项

Windows Mobile 上不支持, 338
数据库服务器, 157

.NET Compact Framework

与 SQL Anywhere for Windows Mobile 一起使用,
311

.odbc.ini

创建数据源, 711
在 DSN 连接参数中指定, 262
在 Mac OS X 上创建数据源, 93
存储加密口令, 265

.saplan 文件

计划查看器文件扩展名, 652

[服务器启动选项] 窗口

在 Linux 上使用, 219

[新建成员关系] 窗口

使用, 444

@data 选项

Interactive SQL [dbisql] 实用程序, 740

校验 [dbvalid] 实用程序, 810

ping [dbping] 实用程序, 757

SQL Anywhere 控制台 [dbconsole] 实用程序,
780

Windows Mobile 上不支持, 338

Windows 服务 [dbsvc] 实用程序, 773

事务日志 [dblog] 实用程序, 792

信息 [dbinfo] 实用程序, 728

停止 [dbstop] 实用程序, 783

关于, 695

初始化 [dbinit] 实用程序, 729

升级 [dbupgrad] 实用程序, 808

卸载 [dbunload] 实用程序, 795

备份 [dbbackup] 实用程序, 698

广播转发器 [dbns11] 实用程序, 703

支持 [dbsupport] 语法, 785

数据库服务器, 156

数据源 [dbdsn] 实用程序, 710

日志传送管理器实用程序 [dbltm] 语法, 748

服务器授权 [dblic] 实用程序, 766

服务器枚举 [dblocate] 实用程序, 763

消除 [dberase] 实用程序, 722

生成 [dbspawn] 实用程序, 781

直方图 [dbhist] 实用程序, 726

翻译日志文件 [dbtran] 实用程序, 753

@environment-variable 选项 (见 @data 选项)

@filename 选项 (见 @data 选项)

&

Unix 命令行, 57

在配置文件中使用时, 695

#

在配置文件中使用时, 695

1254TRKALT 归类

使用, 412

1254TRK 归类

不同于 1254TRKALT, 412

-ac 选项

卸载 [dbunload] 实用程序, 795

-ad 选项

数据库服务器, 234

-af 选项

初始化 [dbinit] 实用程序, 729

-an 选项

卸载 [dbunload] 实用程序, 795

-ap 选项

卸载 [dbunload] 实用程序, 795

广播转发器 [dbns11] 实用程序, 703

-ar 选项

卸载 [dbunload] 实用程序, 795

数据库服务器, 235

-as 选项

Linux 服务 [dbsvc] 实用程序, 770

Windows 服务 [dbsvc] 实用程序, 774

数据库服务器, 236

-a 选项

Linux 服务 [dbsvc] 实用程序, 770

Windows 服务 [dbsvc] 实用程序, 774

初始化 [dbinit] 实用程序, 729

数据库服务器, 234

日志传送管理器 [dbltm] 实用程序, 748

翻译日志文件 [dbtran] 实用程序, 753

-b 选项

初始化 [dbinit] 实用程序, 729

备份 [dbbackup] 实用程序, 698

数据库服务器, 157

数据源 [dbdsn] 实用程序, 711

-ca 选项

数据库服务器, 159

-cc 选项

支持 [dbsupport] 语法, 787

- 数据库服务器, 160
- cd 选项
 - 支持 [dbsupport] 语法, 787
- cet 选项
 - 支持 [dbsupport] 语法, 787
- ce 选项
 - 支持 [dbsupport] 语法, 787
- ch 选项
 - 支持 [dbsupport] 语法, 787
 - 数据库服务器, 161
- cid 选项
 - 支持 [dbsupport] 语法, 787
- cl 选项
 - 数据库服务器, 162
 - 数据源 [dbdsn] 实用程序, 710
- cm 选项
 - Linux 服务 [dbsvc] 实用程序, 771
 - Windows Mobile 上不支持, 338
 - Windows 服务 [dbsvc] 实用程序, 777
 - 卸载 [dbunload] 实用程序, 795
 - 数据库服务器, 163
 - 数据源 [dbdsn] 实用程序, 711
- cp 选项
 - 卸载 [dbunload] 实用程序, 795
 - 支持 [dbsupport] 语法, 787
 - 数据库服务器, 164
- cr 选项
 - 支持 [dbsupport] 语法, 787
 - 数据库服务器, 165
- cs 选项
 - 数据库服务器, 165
- cv 选项
 - 数据库服务器, 166
- cw 选项
 - Windows Mobile 上不支持, 338
 - 数据库服务器, 167
 - 数据源 [dbdsn] 实用程序, 713
- c 选项
 - dbisqlc 实用程序, 720
 - Interactive SQL [dbisql] 实用程序, 740
 - 校验 [dbvalid] 实用程序, 810
 - ping [dbping] 实用程序, 757
 - SQL Anywhere 控制台 [dbconsole] 实用程序, 780
 - SQL Anywhere 脚本执行 [dbrunsql] 实用程序, 761
 - 信息 [dbinfo] 实用程序, 728
 - 停止 [dbstop] 实用程序, 783
 - 初始化 [dbinit] 实用程序, 729
 - 升级 [dbupgrad] 实用程序, 808
 - 卸载 [dbunload] 实用程序, 795
 - 备份 [dbbackup] 实用程序, 698
 - 数据库服务器, 158
 - 数据源 [dbdsn] 实用程序, 713
 - 日志传送管理器 [dbltm] 实用程序, 748
 - 直方图 [dbhist] 实用程序, 726
 - 翻译日志文件 [dbtran] 实用程序, 753
 - 连接字符串, 81
- d1 选项
 - Interactive SQL [dbisql] 实用程序, 740
- datasource 选项
 - Interactive SQL [dbisql] 实用程序, 740
 - SQL Anywhere 控制台 [dbconsole] 实用程序, 780
- dba 选项
 - 初始化 [dbinit] 实用程序, 729
- dbs 选项
 - 初始化 [dbinit] 实用程序, 729
- dc 选项
 - 卸载 [dbunload] 实用程序, 795
- dh 选项
 - 数据库服务器, 237
- dl 选项
 - 日志传送管理器 [dbltm] 实用程序, 748
- dn 选项
 - 服务器枚举 [dblocate] 实用程序, 763
- dr 选项
 - 数据源 [dbdsn] 实用程序, 711
- ds 选项
 - 数据库服务器, 237
- dt 选项
 - 数据库服务器, 170
- dv 选项
 - 服务器枚举 [dblocate] 实用程序, 763
- d 选项
 - dbisqlc 实用程序, 720
 - Interactive SQL [dbisql] 实用程序, 740
 - 校验 [dbvalid] 实用程序, 810
 - Linux 服务 [dbsvc] 实用程序, 769
 - MobiLink [viewcert], 708
 - ping [dbping] 实用程序, 757
 - SQL Anywhere 脚本执行 [dbrunsql] 实用程序, 761
 - Windows 服务 [dbsvc] 实用程序, 773

-
- 停止 [dbstop] 实用程序, 783
 - 卸载 [dbunload] 实用程序, 795
 - 备份 [dbbackup] 实用程序, 698
 - 数据源 [dbdsn] 实用程序, 710
 - 服务器枚举 [dblocate] 实用程序, 763
 - 翻译日志文件 [dbtran] 实用程序, 753
 - ea 选项
 - 初始化 [dbinit] 实用程序, 729
 - 卸载 [dbunload] 实用程序, 795
 - ec 选项
 - 保护客户端/服务器通信安全, 1042
 - 数据库服务器, 171
 - ek 选项
 - 事务日志 [dblog] 实用程序, 792
 - 初始化 [dbinit] 实用程序, 729
 - 卸载 [dbunload] 实用程序, 795
 - 数据库服务器, 238
 - 日志传送管理器 [dbltm] 实用程序, 748
 - 消除 [dberase] 实用程序, 722
 - 翻译日志文件 [dbtran] 实用程序, 753
 - en 选项
 - ping [dbping] 实用程序, 757
 - ep 选项
 - 事务日志 [dblog] 实用程序, 792
 - 初始化 [dbinit] 实用程序, 729
 - 卸载 [dbunload] 实用程序, 795
 - 数据库服务器, 173
 - 日志传送管理器 [dbltm] 实用程序, 748
 - 消除 [dberase] 实用程序, 722
 - 翻译日志文件 [dbtran] 实用程序, 753
 - er 选项
 - 卸载 [dbunload] 实用程序, 795
 - es 选项
 - 数据库服务器, 174
 - et 选项
 - 初始化 [dbinit] 实用程序, 729
 - 卸载 [dbunload] 实用程序, 795
 - e 选项
 - SQL Anywhere 脚本执行 [dbrunsql] 实用程序, 761
 - 卸载 [dbunload] 实用程序, 795
 - 支持 [dbsupport] 语法, 785
 - fc 选项
 - 数据库服务器, 175
 - fips 服务器选项
 - AES256_FIPS 加密算法, 176
 - AES_FIPS 加密算法, 176
 - fx 选项
 - 校验 [dbvalid] 实用程序, 810
 - f 选项
 - Interactive SQL [dbisql] 实用程序, 740
 - SQL Anywhere 脚本执行 [dbrunsql] 实用程序, 761
 - 数据库服务器, 175
 - 数据源 [dbdsn] 实用程序, 711
 - 生成 [dbspawn] 实用程序, 781
 - 翻译日志文件 [dbtran] 实用程序, 753
 - ga 选项
 - 数据库服务器, 177
 - gb 选项
 - Windows Mobile 上不支持, 338
 - 数据库服务器, 178
 - gc 选项
 - 数据库服务器, 178
 - gd 选项
 - 数据库服务器, 179
 - ge 选项
 - Windows Mobile 上不支持, 338
 - 数据库服务器, 180
 - gf 选项
 - 数据库服务器, 180
 - gk 选项
 - 数据库服务器, 181
 - gl 选项
 - 数据库服务器, 181
 - gm 选项
 - 数据库服务器, 182
 - gn 选项
 - 影响查询内并行机制, 182
 - 数据库服务器, 182
 - 数据库服务器使用, 48
 - 数据库服务器进程并发水平, 50
 - gp 选项
 - 数据库服务器, 183
 - gr 选项
 - 数据库服务器, 184
 - gss 选项
 - 数据库服务器, 185
 - 数据库服务器使用, 48
 - gtc 选项
 - 数据库服务器, 186
 - 数据库服务器使用, 49
 - gt 选项
 - 数据库服务器, 185

- 数据库服务器使用, 48
- gu 选项
 - 控制实用程序数据库的语句执行权限, 30
 - 数据库服务器, 188
- g 选项
 - Linux 服务 [dbsvc] 实用程序, 769
 - SQL Anywhere 脚本执行 [dbrunsql] 实用程序, 761
 - Windows 服务 [dbsvc] 实用程序, 773
 - 事务日志 [dblog] 实用程序, 792
 - 卸载 [dbunload] 实用程序, 795
 - 数据源 [dbdsn] 实用程序, 710
 - 翻译日志文件 [dbtran] 实用程序, 753
- host 选项
 - Interactive SQL [dbisql] 实用程序, 740
 - SQL Anywhere 控制台 [dbconsole] 实用程序, 780
- ii 选项
 - 卸载 [dbunload] 实用程序, 795
- il 选项
 - 事务日志 [dblog] 实用程序, 792
- im 选项
 - 内存模式, 46
 - 数据库服务器, 188
- ip 选项
 - MobiLink [viewcert], 708
- ir 选项
 - 事务日志 [dblog] 实用程序, 792
 - 翻译日志文件 [dbtran] 实用程序, 753
- is 选项
 - 事务日志 [dblog] 实用程序, 792
 - 支持 [dbsupport] 语法, 785
 - 翻译日志文件 [dbtran] 实用程序, 753
- it 选项
 - 翻译日志文件 [dbtran] 实用程序, 753
- iu 选项
 - 支持 [dbsupport] 语法, 785
- ix 选项
 - 卸载 [dbunload] 实用程序, 795
- i 选项
 - 校验 [dbvalid] 实用程序, 810
 - Linux 服务 [dbsvc] 实用程序, 769
 - Windows 服务 [dbsvc] 实用程序, 774
 - 初始化 [dbinit] 实用程序, 729
 - 升级 [dbupgrad] 实用程序, 808
 - 日志传送管理器 [dbltm] 实用程序, 748
- j 选项
 - 翻译日志文件 [dbtran] 实用程序, 753
- kl 选项
 - 数据库服务器, 190
- krb 选项
 - 数据库服务器, 192
- kr 选项
 - 数据库服务器, 191
- ksc 选项
 - 数据库服务器, 193
- ksd 选项
 - 数据库服务器, 193
- ks 选项
 - 数据库服务器, 193
- k 选项
 - 初始化 [dbinit] 实用程序, 729
 - 卸载 [dbunload] 实用程序, 795
 - 备份 [dbbackup] 实用程序, 698
 - 数据库服务器, 190
 - 翻译日志文件 [dbtran] 实用程序, 753
- lc 选项
 - 支持 [dbsupport] 语法, 785
- le 选项
 - 初始化 [dbinit] 实用程序, 729
- ls 选项
 - 支持 [dbsupport] 语法, 785
- l 选项
 - ping [dbping] 实用程序, 757
 - Windows 服务 [dbsvc] 实用程序, 773
 - 初始化 [dbinit] 实用程序, 729
 - 卸载 [dbunload] 实用程序, 795
 - 备份 [dbbackup] 实用程序, 698
 - 数据源 [dbdsn] 实用程序, 710
 - 服务器授权 [dblic] 实用程序, 766
- m 选项
 - ping [dbping] 实用程序, 757
 - 事务日志 [dblog] 实用程序, 792
 - 初始化 [dbinit] 实用程序, 729
 - 卸载 [dbunload] 实用程序, 795
 - 广播转发器 [dbns11] 实用程序, 703
 - 数据库服务器, 194, 238
 - 日志传送管理器 [dbltm] 实用程序, 748
 - 翻译日志文件 [dbtran] 实用程序, 753
 - 语言 [dblang] 实用程序, 745
- nl 选项
 - 卸载 [dbunload] 实用程序, 795
- nogui 选项
 - Interactive SQL [dbisql] 实用程序, 740

-
- no 选项
 - 卸载 [dbunload] 实用程序, 795
 - nr 选项
 - 支持 [dbsupport] 语法, 790
 - ns 选项
 - 数据源 [dbdsn] 实用程序, 711
 - n 选项
 - 事务日志 [dblog] 实用程序, 792
 - 初始化 [dbinit] 实用程序, 729
 - 卸载 [dbunload] 实用程序, 795
 - 备份 [dbbackup] 实用程序, 698
 - 数据库选项, 239
 - 服务器枚举 [dblocate] 实用程序, 763
 - 服务器选项, 195
 - 直方图 [dbhist] 实用程序, 726
 - 翻译日志文件 [dbtran] 实用程序, 753
 - 设置数据库名, 239
 - od 选项
 - Linux 服务 [dbsvc] 实用程序, 770
 - oe 选项
 - 安静地运行, 46
 - 记录启动错误, 197
 - onerror 选项
 - Interactive SQL [dbisql] 实用程序, 740
 - on 选项
 - 数据库服务器, 198
 - op 选项
 - MobiLink [viewcert], 708
 - or 选项
 - 数据源 [dbdsn] 实用程序, 711
 - os 选项
 - 数据库服务器, 198
 - 日志传送管理器 [dbltn] 实用程序, 748
 - ot 选项
 - 数据库服务器, 199
 - 日志传送管理器 [dbltn] 实用程序, 748
 - o 选项
 - 校验 [dbvalid] 实用程序, 810
 - MobiLink [viewcert], 708
 - ping [dbping] 实用程序, 757
 - SQL Anywhere 脚本执行 [dbrunsql] 实用程序, 761
 - Windows 服务 [dbsvc] 实用程序, 777
 - 事务日志 [dblog] 实用程序, 792
 - 信息 [dbinfo] 实用程序, 728
 - 停止 [dbstop] 实用程序, 783
 - 初始化 [dbinit] 实用程序, 729
 - 升级 [dbupgrad] 实用程序, 808
 - 卸载 [dbunload] 实用程序, 795
 - 备份 [dbbackup] 实用程序, 698
 - 安静地运行, 46
 - 广播转发器 [dbns11] 实用程序, 703
 - 支持 [dbsupport] 语法, 785
 - 数据库服务器, 196
 - 数据源 [dbdsn] 实用程序, 711
 - 日志传送管理器 [dbltn] 实用程序, 748
 - 服务器授权 [dblic] 实用程序, 766
 - 服务器枚举 [dblocate] 实用程序, 763
 - 消除 [dberase] 实用程序, 722
 - 翻译日志文件 [dbtran] 实用程序, 753
 - pc 选项
 - ping [dbping] 实用程序, 757
 - 支持 [dbsupport] 语法, 785
 - 数据库服务器, 200
 - pd 选项
 - ping [dbping] 实用程序, 757
 - 支持 [dbsupport] 语法, 785
 - pe 选项
 - 数据源 [dbdsn] 实用程序, 711
 - port 选项
 - Interactive SQL [dbisql] 实用程序, 740
 - SQL Anywhere 控制台 [dbconsole] 实用程序, 780
 - pr 选项
 - Linux 服务 [dbsvc] 实用程序, 770
 - ps 选项
 - ping [dbping] 实用程序, 757
 - 支持 [dbsupport] 语法, 785
 - pt 选项
 - 数据库服务器, 201
 - p 选项
 - MobiLink [viewcert], 708
 - Windows 服务 [dbsvc] 实用程序, 774
 - 初始化 [dbinit] 实用程序, 729
 - 卸载 [dbunload] 实用程序, 795
 - 广播转发器 [dbns11] 实用程序, 703
 - 数据库服务器, 200
 - 服务器枚举 [dblocate] 实用程序, 763
 - 生成 [dbspawn] 实用程序, 781
 - qc 选项
 - SQL Anywhere 脚本执行 [dbrunsql] 实用程序, 761
 - 卸载 [dbunload] 实用程序, 795
 - qi 选项
-

- Windows Mobile 上不支持, 338
- 安静地运行, 46
- 数据库服务器, 201
- qn 选项
 - 数据库服务器, 202
- qp 选项
 - 数据库服务器, 203
- qs 选项
 - 安静地运行, 46
 - 数据库服务器, 203
- qw 选项
 - 安静地运行, 46
 - 数据库服务器, 204
- q 选项
 - dbisqlc 实用程序, 720
 - Interactive SQL [dbisql] 实用程序, 740
 - 校验 [dbvalid] 实用程序, 810
 - Linux 服务 [dbsvc] 实用程序, 771
 - ping [dbping] 实用程序, 757
 - SQL Anywhere 脚本执行 [dbrunsql] 实用程序, 761
 - Windows 服务 [dbsvc] 实用程序, 777
 - 事务日志 [dblog] 实用程序, 792
 - 信息 [dbinfo] 实用程序, 728
 - 停止 [dbstop] 实用程序, 783
 - 初始化 [dbinit] 实用程序, 729
 - 升级 [dbupgrad] 实用程序, 808
 - 卸载 [dbunload] 实用程序, 795
 - 备份 [dbbackup] 实用程序, 698
 - 广播转发器 [dbns11] 实用程序, 703
 - 支持 [dbsupport] 语法, 785
 - 数据源 [dbdsn] 实用程序, 711
 - 日志传送管理器 [dbltm] 实用程序, 748
 - 服务器授权 [dblic] 实用程序, 766
 - 服务器枚举 [dblocate] 实用程序, 763
 - 消除 [dberase] 实用程序, 722
 - 生成 [dbspawn] 实用程序, 781
 - 翻译日志文件 [dbtran] 实用程序, 753
 - 语言 [dblangu] 实用程序, 745
- rd 选项
 - 支持 [dbsupport] 语法, 790
- rg 选项
 - Windows 服务 [dbsvc] 实用程序, 774
- rr 选项
 - 支持 [dbsupport] 语法, 790
- rsu 选项
 - 翻译日志文件 [dbtran] 实用程序, 753
- rs 选项
 - Linux 服务 [dbsvc] 实用程序, 770
 - Windows 服务 [dbsvc] 实用程序, 774
- r 选项
 - Linux 服务 [dbsvc] 实用程序, 770
 - MobiLink [createcert], 705
 - 事务日志 [dblog] 实用程序, 792
 - 卸载 [dbunload] 实用程序, 795
 - 备份 [dbbackup] 实用程序, 698
 - 支持 [dbsupport] 语法, 790
 - 数据库, 240
 - 数据库服务器, 204
 - 翻译日志文件 [dbtran] 实用程序, 753
- sa 选项
 - 支持 [dbsupport] 语法, 785
- sb 选项
 - 数据库服务器, 206
- sc 选项
 - 支持 [dbsupport] 语法, 785
- sd 选项
 - Windows 服务 [dbsvc] 实用程序, 774
 - 支持 [dbsupport] 语法, 785
- sf 选项
 - 数据库服务器, 207
- sk 选项
 - 数据库服务器, 210
- sm 选项
 - 数据库, 241
 - 用于访问镜像数据库, 898
- sn 选项
 - Windows 服务 [dbsvc] 实用程序, 774
 - 数据库, 242
- sr 选项
 - 翻译日志文件 [dbtran] 实用程序, 753
- ss 选项
 - 服务器枚举 [dblocate] 实用程序, 763
- status 选项
 - Linux 服务 [dbsvc] 实用程序, 770
- st 选项
 - ping [dbping] 实用程序, 757
- su 选项
 - 数据库服务器, 211
 - 连接到实用程序数据库, 29
- s 选项
 - 校验 [dbvalid] 实用程序, 810
 - Linux 服务 [dbsvc] 实用程序, 770
 - MobiLink [createcert], 705

- ping [dbping] 实用程序, 757
- SQL Anywhere 脚本执行 [dbrunsql] 实用程序, 761
- Windows Mobile 上不支持, 338
- Windows 服务 [dbsvc] 实用程序, 774
- 初始化 [dbinit] 实用程序, 729
- 备份 [dbbackup] 实用程序, 698
- 广播转发器 [dbns11] 实用程序, 703
- 数据库服务器, 205
- 日志传送管理器 [dbltm] 实用程序, 748
- 服务器枚举 [dblocate] 实用程序, 763
- 翻译日志文件 [dbtran] 实用程序, 753
- ti 选项
 - 数据库服务器, 212
- tl 选项
 - 数据库服务器, 212
- tmf 选项
 - Windows Mobile 上不支持, 338
 - 数据库服务器, 213
- tmt 选项
 - Windows Mobile 上不支持, 338
 - 数据库服务器, 214
- tq 时间选项
 - 数据库服务器, 214
- t 选项
 - 校验 [dbvalid] 实用程序, 810
 - Linux 服务 [dbsvc] 实用程序, 770
 - Windows 服务 [dbsvc] 实用程序, 773, 774
 - 事务日志 [dblog] 实用程序, 792
 - 初始化 [dbinit] 实用程序, 729
 - 卸载 [dbunload] 实用程序, 795
 - 备份 [dbbackup] 实用程序, 698
 - 直方图 [dbhist] 实用程序, 726
 - 翻译日志文件 [dbtran] 实用程序, 753
- ua 选项
 - Windows Mobile 上不支持, 338
 - 数据库服务器, 215
- uc 选项
 - Windows Mobile 上不支持, 338
 - 数据库服务器, 215
- ud 选项
 - Windows Mobile 上不支持, 338
 - 数据库服务器, 216
 - 日志传送管理器 [dbltm] 实用程序, 748
- uf 选项
 - Windows Mobile 上不支持, 338
 - 数据库服务器, 217
- ui 选项
 - Windows Mobile 上不支持, 338
 - 数据库服务器, 217
- ul 选项
 - Interactive SQL [dbisql] 实用程序, 740
- um 选项
 - 数据库服务器, 218
- ut 选项
 - Windows Mobile 上不支持, 338
 - 数据库服务器, 218
- ux 选项
 - Windows Mobile 上不支持, 338
 - 数据库服务器, 219
 - 日志传送管理器 [dbltm] 实用程序, 748
- u 选项
 - Linux 服务 [dbsvc] 实用程序, 769
 - Windows Mobile 上不支持, 338
 - Windows 服务 [dbsvc] 实用程序, 773
 - 信息 [dbinfo] 实用程序, 728
 - 卸载 [dbunload] 实用程序, 795
 - 数据库服务器, 215
 - 服务器授权 [dblic] 实用程序, 766
 - 直方图 [dbhist] 实用程序, 726
 - 翻译日志文件 [dbtran] 实用程序, 753
 - 语言 [dblang] 实用程序, 745
- version
 - Interactive SQL [dbisql] 实用程序, 740
- vss 选项
 - 数据库服务器, 220
- v 选项
 - SQL Anywhere 脚本执行 [dbrunsql] 实用程序, 761
 - 卸载 [dbunload] 实用程序, 795
 - 数据库服务器, 220
 - 数据源 [dbdsn] 实用程序, 711
 - 日志传送管理器 [dbltm] 实用程序, 748
 - 服务器枚举 [dblocate] 实用程序, 763
- w 选项
 - Linux 服务 [dbsvc] 实用程序, 769
 - Windows 服务 [dbsvc] 实用程序, 773
 - 数据源 [dbdsn] 实用程序, 710
- xa 选项
 - 数据库服务器, 222
- xd 选项
 - 数据库服务器, 223
- xf 选项
 - 数据库服务器, 223

- xi 选项
 - 卸载 [dbunload] 实用程序, 795
- xo 选项
 - 备份 [dbbackup] 实用程序, 698
- xp 选项
 - Windows Mobile 上不支持, 338
 - 数据库, 244
- xs 选项
 - 保护通信安全, 1046
 - 数据库服务器, 224
- xx 选项
 - 卸载 [dbunload] 实用程序, 795
- x 选项
 - dbisqlc 实用程序, 720
 - Interactive SQL [dbisql] 实用程序, 740
 - Linux 服务 [dbsvc] 实用程序, 769
 - Windows 服务 [dbsvc] 实用程序, 773
 - 事务日志 [dblog] 实用程序, 792
 - 停止 [dbstop] 实用程序, 783
 - 备份 [dbbackup] 实用程序, 698
 - 广播转发器 [dbns11] 实用程序, 703
 - 数据库服务器, 221
 - 翻译日志文件 [dbtran] 实用程序, 753
- y 选项
 - Linux 服务 [dbsvc] 实用程序, 771
 - Windows 服务 [dbsvc] 实用程序, 777
 - 停止 [dbstop] 实用程序, 783
 - 卸载 [dbunload] 实用程序, 795
 - 备份 [dbbackup] 实用程序, 698
 - 数据源 [dbdsn] 实用程序, 711
 - 消除 [dberase] 实用程序, 722
 - 翻译日志文件 [dbtran] 实用程序, 753
- ze 选项
 - Windows Mobile 上不支持, 338
 - 初始化 [dbinit] 实用程序, 729
 - 数据库服务器, 226
- zl 选项
 - 数据库服务器, 227
- zn 选项
 - 初始化 [dbinit] 实用程序, 729
 - 数据库服务器, 228
- zoc 选项
 - 数据库服务器, 229
- zo 选项
 - 数据库服务器, 229
- zp 选项
 - 数据库服务器, 230

- zr 选项
 - 数据库服务器, 230
- zs 选项
 - 数据库服务器, 232
- zt 选项
 - 数据库服务器, 233
- z 选项
 - ping [dbping] 实用程序, 757
 - 调试网络通信问题, 68
 - 事务日志 [dblog] 实用程序, 792
 - 初始化 [dbinit] 实用程序, 729
 - 广播转发器 [dbns11] 实用程序, 703
 - 数据库服务器, 226
 - 翻译日志文件 [dbtran] 实用程序, 753

A

- AccentSensitive 属性
 - SQL Anywhere SNMP Extension Agent OID, 984
 - 数据库属性说明, 601
- ActiveReq 属性
 - SQL Anywhere SNMP Extension Agent OID, 972
 - 服务器属性说明, 588
- ActiveSync
 - SQL Anywhere for Windows Mobile 需要的版本, 310
 - Vista, 39
- ActiveSync 提供程序的安装实用程序 (mlasinst)
 - 在 Vista 上可能需要权限提升, 38
- Address Windowing Extensions
 - 限制高速缓存大小, 167
- ADO
 - 连接, 97
- ADO.NET 示例
 - 使用, 314
- AES_FIPS 加密算法
 - fips 服务器选项, 176
 - 初始化 [dbinit] 实用程序, 729
 - 卸载 [dbunload] 实用程序, 795
- AES256_FIPS 加密算法
 - fips 服务器选项, 176
 - 初始化 [dbinit] 实用程序, 729
 - 卸载 [dbunload] 实用程序, 795
- AES256 加密算法
 - 初始化 [dbinit] 实用程序, 729
 - 卸载 [dbunload] 实用程序, 795
- AES 加密算法
 - 关于, 1019

-
- 初始化 [dbinit] 实用程序, 729
 - 卸载 [dbunload] 实用程序, 795
 - Agent 表
 - SQL Anywhere MIB, 969
 - AIX
 - IPv6 支持, 135
 - LIBPATH 环境变量, 348
 - 使用 LDAP 服务器, 138
 - Alias 属性
 - SQL Anywhere SNMP Extension Agent OID, 984
 - 数据库属性说明, 601
 - allow_nulls_by_default 选项
 - ASE 兼容性, 473
 - Open Client, 1067
 - SQL Anywhere SNMP Extension Agent OID, 987
 - Transact-SQL 兼容性, 473
 - 说明, 477
 - 连接属性说明, 566
 - allow_read_client_file 选项
 - SQL Anywhere SNMP Extension Agent OID, 987
 - 说明, 478
 - 连接属性说明, 566
 - allow_snapshot_isolation 选项
 - SQL Anywhere SNMP Extension Agent OID, 987
 - 说明, 478
 - 连接属性说明, 566
 - allow_write_client_file 选项
 - SQL Anywhere SNMP Extension Agent OID, 987
 - 说明, 479
 - 连接属性说明, 566
 - ALL 权限
 - 关于, 428
 - 授予, 434
 - ALTER DATABASE 语句
 - Windows Mobile 上的限制, 337
 - 关闭主服务器, 899
 - 在镜像系统中强制故障转移, 899
 - ALTER LOGIN POLICY 语句
 - 修改登录策略, 419
 - AlternateMirrorServerName 属性
 - SQL Anywhere SNMP Extension Agent OID, 984
 - 数据库属性说明, 601
 - AlternateServerName 属性
 - SQL Anywhere SNMP Extension Agent OID, 984
 - 数据库属性说明, 601
 - ALTER PROCEDURE 语句
 - 设置 REPLICATE ON 的效果, 1086
 - ALTER TABLE 语句
 - REPLICATE ON, 1076
 - ALTER USER 语句
 - 为现有用户分配登录策略, 418
 - 口令, 432
 - ALTER 权限
 - 关于, 428
 - 授予, 434
 - ANSI
 - cooperative_commits 选项, 494
 - delayed_commits 选项, 501
 - 一致性, 545
 - 关于代码页, 386
 - 删除权限, 481
 - 变量行为, 480
 - 数据库兼容性选项, 472
 - 更新权限, 481
 - 游标, 481
 - ansi_blanks 选项
 - ASE 兼容性, 473
 - Open Client, 1067
 - SQL Anywhere SNMP Extension Agent OID, 987
 - Transact-SQL 兼容性, 473
 - 说明, 480
 - 连接属性说明, 566
 - ansi_close_cursors_on_rollback 选项
 - SQL Anywhere SNMP Extension Agent OID, 987
 - Transact-SQL 兼容性, 473
 - 说明, 481
 - 连接属性说明, 566
 - ansi_permissions 选项
 - SQL Anywhere SNMP Extension Agent OID, 987
 - Transact-SQL 兼容性, 473
 - 说明, 481
 - 连接属性说明, 566
 - ansi_substring 选项
 - ASE 兼容性, 473
 - SQL Anywhere SNMP Extension Agent OID, 987
 - 说明, 482
 - 连接属性说明, 566
 - ansi_update_constraints 选项
 - SQL Anywhere SNMP Extension Agent OID, 987
 - Transact-SQL 兼容性, 473
 - 说明, 483
 - 连接属性说明, 566
 - ansinull 选项
 - ASE 兼容性, 473
-

- Open Client, 1067
- SQL Anywhere SNMP Extension Agent OID, 987
- Transact-SQL 兼容性, 473
- 说明, 484
- 连接属性说明, 566
- APC
 - 关于, 1086
 - 函数 APC, 1085
 - 复制服务器, 1071
- APC_pw 参数
 - LTM 配置文件, 750
 - 启动 LTM, 1079
- APC_user 参数
 - LTM 配置文件, 750
 - 关于, 1086
 - 启动 LTM, 1079
- API
 - 通过 SQL Anywhere 连接, 83
- AppInfo 连接参数
 - 说明, 249
- AppInfo 属性
 - 连接属性说明, 566
- ApproximateCPUTime 属性
 - 连接属性说明, 566
- APP 连接参数
 - 说明, 249
- ArbiterState 属性
 - SQL Anywhere SNMP Extension Agent OID, 984
 - 数据库属性说明, 601
- ASCII
 - 字符集, 385
- ASE
 - (参见 Adaptive Server Enterprise)
 - 替代字符集编码标签, 404
- ASTART 连接参数
 - 说明, 250
- ASTOP 连接参数
 - 说明, 251
- auditing
 - 连接属性说明, 566
- auditing_options 选项
 - SQL Anywhere SNMP Extension Agent OID, 987
 - 说明, 485
 - 连接属性说明, 566
- AuditingTypes 属性
 - SQL Anywhere SNMP Extension Agent OID, 984
 - 数据库属性说明, 601
- auditing 选项
 - SQL Anywhere SNMP Extension Agent OID, 987
 - 说明, 484
 - 连接属性说明, 566
- authdn 参数
 - LDAP, 139
- Authenticated 属性
 - 数据库属性说明, 601
 - 连接属性说明, 566
- AuthType 属性
 - 连接属性说明, 566
- auto_commit 选项
 - Interactive SQL 设置, 667
 - 说明, 668
- auto_refetch 选项
 - Interactive SQL 设置, 667
 - 说明, 669
- automatic_timestamp 选项
 - Transact-SQL 兼容性, 473
- AutoStart 连接参数
 - 说明, 250
- AutoStop 连接参数
 - 说明, 251
- AvailIO 属性
 - SQL Anywhere SNMP Extension Agent OID, 972
 - 服务器属性说明, 588
- AWE 高速缓存
 - cm 服务器选项, 163
 - cw 服务器选项, 167
 - 在 Vista 上运行 SQL Anywhere, 39
- 安静模式
 - dbisqlc 实用程序, 720
 - Interactive SQL [dbisql] 实用程序, 740
 - 校验 [dbvalid] 实用程序, 810
 - ping [dbping] 实用程序, 757
 - SQL Anywhere 脚本执行 [dbrunsql] 实用程序, 761
 - 事务日志 [dblog] 实用程序, 792
 - 信息 [dbinfo] 实用程序, 728
 - 停止 [dbstop] 实用程序, 783
 - 初始化 [dbinit] 实用程序, 729
 - 升级 [dbupgrad] 实用程序, 808
 - 卸载 [dbunload] 实用程序, 795
 - 备份 [dbbackup] 实用程序, 698
 - 数据库服务器, 46
 - 数据源 [dbdsn] 实用程序, 711
 - 日志传送管理器 [dbltm] 实用程序, 748

服务器授权 [dblic] 实用程序, 766
服务器枚举 [dblocate] 实用程序, 763
消除 [dberase] 实用程序, 722
生成 [dbspawn] 实用程序, 781
翻译日志文件 [dbtran] 实用程序, 753
语言 [dblang] 实用程序, 745

安全

- AES 加密, 1019
- auditing 选项, 484
- FIPS, 1033
- Windows Mobile, 1029
- 临时文件, 357
- 事件示例, 870
- 关于, 1003
- 关于传送层安全, 1031
- 创建数据库, 1018
- 删除数据库, 1018
- 卸载数据, 1018
- 口令, 1006
- 复制数据库文件, 115
- 实用程序数据库, 30
- 审计, 1012
- 审计检索, 1013
- 对数据库文件加密, 1019
- 对配置文件进行简单加密, 724
- 提示, 1005
- 数据库服务器, 1005, 1018
- 文件访问, 181
- 文件隐藏 [dbfhide] 实用程序, 724
- 最小口令长度, 522
- 服务, 63
- 服务器命令行, 1004
- 概述, 1004
- 监控器用户, 944
- 禁用数据库功能, 544, 1010
- 系统函数, 1005
- 装载数据, 1018
- 视图, 452
- 集成登录, 114, 1006

安全套接字层

- 关于, 1031

安全性

- ec 服务器选项, 171
- ek 服务器选项, 238
- ep 服务器选项, 173
- es 服务器选项, 174
- DatabaseKey [DBKEY] 连接参数, 259

Encryption [ENC] 连接参数, 265
在 Vista 上运行 SQL Anywhere, 38
过程, 437, 452

安装

- Windows Mobile, 368
- 位置, 368
- 单独计算机上的监控器, 951
- 在 Windows Mobile 上安装 SQL Anywhere, 310
- 注册表设置, 374

安装的对象

- 监控器重新安装, 940

安装脚本

- 关于, 1081
- 准备运行, 1081
- 运行, 1082

安装目录

- 关于, 368

安装要求

- SQL Anywhere for Windows Mobile, 310

安装注意事项

- Windows Mobile, 310, 311

B

background_priority 选项

- SQL Anywhere SNMP Extension Agent OID, 987
- 说明 [不建议使用], 485
- 连接属性说明 [不建议使用], 566

backup.syb 文件

- 确定位置, 356

BackupEnd 系统事件

- 说明, 868

BACKUP 权限

- 联机备份需要, 817

BACKUP 特权

- 不可继承, 424
- 关于, 424
- 授予, 433

BACKUP 语句

- Windows Mobile 上的限制, 337
- 创建映像备份, 826
- 创建档案备份, 826

basedn 参数

- LDAP, 139

batch_ltl_cmds 参数

- LTM 配置文件, 750

batch_ltl_mem 参数

- LTM 配置文件, 750

- batch_ltl_sz 参数
 - LTM 配置文件, 750
- BCAST 协议选项
 - 使用 IPv6 地址, 135
 - 说明, 285
- bell 选项
 - Interactive SQL 设置, 667
 - 说明, 669
- BINARY 数据类型
 - 最大大小, 614
- BlankPadding 属性
 - SQL Anywhere SNMP Extension Agent OID, 984
 - 数据库属性说明, 601
- BLISTENER 协议选项
 - 说明, 286
- blob_threshold 选项
 - SQL Remote 复制选项, 476
 - 说明, 486
- BlockedOn 属性
 - 连接属性说明, 566
- blocking_timeout 选项
 - SQL Anywhere SNMP Extension Agent OID, 987
 - 说明, 487
 - 连接属性说明, 566
- blocking 选项
 - SQL Anywhere SNMP Extension Agent OID, 987
 - 说明, 486
 - 连接属性说明, 566
- BroadcastListener 协议选项
 - 说明, 286
- Broadcast 协议选项
 - 使用 IPv6 地址, 135
 - 说明, 285
- BuildChange 属性
 - 服务器属性说明, 588
- BuildClient 属性
 - 服务器属性说明, 588
- BuildProduction 属性
 - 服务器属性说明, 588
- BuildReproducible 属性
 - 服务器属性说明, 588
- BytesReceivedUncomp 属性
 - SQL Anywhere SNMP Extension Agent OID, 972
 - 服务器属性说明, 588
 - 连接属性说明, 566
- BytesReceived 属性
 - SQL Anywhere SNMP Extension Agent OID, 972
 - 服务器属性说明, 588
 - 连接属性说明, 566
- BytesSentUncomp 属性
 - SQL Anywhere SNMP Extension Agent OID, 972
 - 服务器属性说明, 588
 - 连接属性说明, 566
- BytesSent 属性
 - SQL Anywhere SNMP Extension Agent OID, 972
 - 服务器属性说明, 588
 - 连接属性说明, 566
- 版
 - SQL Anywhere OEM 版, 70
 - SQL Anywhere Web 版, 75
- 版本
 - 数据库服务器, 220
 - 确定, 813
- 版本不匹配
 - 文件位置, 370
- 版本诊断 [dbversion]
 - 语法, 813
- 帮助
 - 技术支持, xvi
- 包
 - 术语定义, 1095
- 包大小
 - 限制, 200, 201
- 保持活动状态的请求标头字段
 - 设置 KeepaliveTimeout 值, 295
- 保护数据的安全
 - 概述, 1003
- 报告
 - 维护计划, 845
- 报告错误
 - 关于, 76
- 备份, 830
 - 重命名备份事务日志, 849
 - 重命名并启动新的事务日志, 698
 - 重命名事务日志, 848
- dbltm, 847
- dbmlsync, 847
- dbremote, 847
- 调度, 844
- 校验, 832, 858
- LTM 管理, 1090
- MobiLink SQL Anywhere 远程数据库, 847
- MobiLink 统一数据库, 853
- 事务日志 - 重命名, 849

- SQL Remote, 847
- Sybase Central, 827
- Windows Mobile 数据库, 328
 - 不包括在复制中的数据库, 853
 - 事务日志 - 使用原始日志, 825
 - 从映像恢复, 835
 - 使用 BACKUP 语句, 825
 - 使用 delete_old_logs 的复制服务器, 1090
 - 关于, 817
 - 内部, 852
 - 删除事务日志, 850
 - 到磁带驱动器, 826
 - 只数据库, 698
 - 在 Windows Mobile 上执行, 328
 - 备份 [dbbackup] 实用程序, 698
 - 复制代理, 847
 - 完全, 820
 - 并行, 855
 - 当前运行的数据库, 817
 - 快速入门, 818
 - 执行权限, 424
 - 数据库镜像, 902
 - 未完成的, 17
 - 档案, 823
 - 比较类型, 819
 - 活动, 821
 - 生成数据库文档, 635
 - 简介, 817
 - 组件, 852
 - 联机, 817
 - 脱机, 817
 - 自动, 844
 - 计划, 844
 - 远程数据库, 847
 - 选择格式, 823
 - 选项, 698
 - 限制, 824
- 备份格式
 - 类型, 823
- 备份和数据恢复
 - 概述, 817
 - 针对 Windows Mobile 的策略, 328
- 备份计划
 - 关于, 843
 - 记录数据库, 635
- 备份目录
 - 备份 [dbbackup] 实用程序, 698
- 备份实用程序 [dbbackup]
 - 客户端备份, 830
 - 接收错误, 698
 - 语法, 698
 - 退出代码, 702
- 备份数据库向导
 - Windows Mobile 上不支持, 339
 - 使用, 828
- 被引用对象
 - 术语定义, 1095
- 本地服务器 (见 个人服务器)
- 本地计算机
 - 环境设置, 373
- 本国语言支持
 - 关于, 378
 - 多字节字符集, 392
- 崩溃
 - 报告, 76
- 比较
 - TIMESTAMP, 500
 - 备份类型, 819
- 编程接口
 - 连接, 83
- 编辑
 - Interactive SQL 中的结果集, 654
 - Interactive SQL 中的表值, 654
 - 连接配置文件, 89
- 编码
 - PKI 对象, 708
 - 字符集, 384
 - 定义, 384
 - 术语定义, 1095
- 编码器
 - 传送层安全, 1031
- 变更登录策略
 - 关于, 419
- 标签
 - 字符集, 408
 - 语言标签值, 389
- 标识
 - 客户端应用程序, 249
- 标识符
 - SQL Anywhere 中的最大长度, 614
 - 不区分大小写, 388
 - 国际方面, 388
 - 复制代理, 1083
 - 术语定义, 1095

标识文件
 关于, 1042
标识协议选项
 dbeng11 -ec, 171
 dbsrv11 -ec, 171
表
 从 Sybase Central 校验, 860
 RESOURCE 特权, 427
 加密, 1026, 1027
 启用表加密, 1027
 在 Interactive SQL 中查找, 648
 复制, 1076, 1084
 所有者, 429
 权限, 428
 确定表占用的磁盘空间, 728
 组所有者, 447
 解密, 1026
 限制, 614
 限定名, 450
表大小
 行数, 614
 限制, 614
表格式数据流通信协议
 Open Server, 1058
表加密
 关于, 1026
 初始化 [dbinit] 实用程序, 729
表名
 国际方面, 388
 在由组拥有时进行限定, 447
表权限
 设置, 434
表值
 在 Interactive SQL 中编辑, 654
并发
 术语定义, 1095
并发连接
 设置最大值, 182
并行备份
 dbbackup 实用程序, 698
 Windows Mobile 上不支持, 336
 关于, 855
并行操作
 max_query_tasks 选项, 519
并行执行
 处理器, 185
拨号网络

 连接, 137
不支持的功能
 Windows Mobile 上的 SQL Anywhere 限制, 336
布尔值
 协议选项, 284
 连接参数, 248
布线
 故障排除, 144
部署
 Vista 注意事项, 39
部署软件
 支持的语言, 379

C

CacheAllocated 属性
 SQL Anywhere SNMP Extension Agent OID, 972
 服务器属性说明, 588
CacheFileDirty 属性
 SQL Anywhere SNMP Extension Agent OID, 972
 服务器属性说明, 588
CacheFile 属性
 SQL Anywhere SNMP Extension Agent OID, 972
 服务器属性说明, 588
CacheFree 属性
 SQL Anywhere SNMP Extension Agent OID, 972
 服务器属性说明, 588
CacheHits 属性
 SQL Anywhere SNMP Extension Agent OID, 972, 980
 数据库属性说明, 601
 服务器属性说明, 588
 连接属性说明, 566
CachePanics 属性
 SQL Anywhere SNMP Extension Agent OID, 972
 服务器属性说明, 588
CachePinned 属性
 SQL Anywhere SNMP Extension Agent OID, 972
 服务器属性说明, 588
CacheReadIndInt 属性
 SQL Anywhere SNMP Extension Agent OID, 980
 数据库属性说明, 601
 连接属性说明, 566
CacheReadIndLeaf 属性
 SQL Anywhere SNMP Extension Agent OID, 980
 数据库属性说明, 601
 连接属性说明, 566
CacheReadTable 属性

- SQL Anywhere SNMP Extension Agent OID, 980
 - 数据库属性说明, 601
 - 连接属性说明, 566
- CacheReadWorkTable 属性
 - SQL Anywhere SNMP Extension Agent OID, 980
 - 数据库属性说明, 601
 - 连接属性说明, 566
- CacheRead 属性
 - SQL Anywhere SNMP Extension Agent OID, 972, 980
 - 数据库属性说明, 601
 - 服务器属性说明, 588
 - 连接属性说明, 566
- CacheReplacements 属性
 - SQL Anywhere SNMP Extension Agent OID, 972
 - 服务器属性说明, 588
- CacheScavenges 属性
 - SQL Anywhere SNMP Extension Agent OID, 972
 - 服务器属性说明, 588
- CacheScavengeVisited 属性
 - SQL Anywhere SNMP Extension Agent OID, 972
 - 服务器属性说明, 588
- CacheSizingStatistics 属性
 - SQL Anywhere SNMP Extension Agent OID, 975
 - 服务器属性说明, 588
- Capabilities 属性
 - SQL Anywhere SNMP Extension Agent OID, 984
 - 数据库属性说明, 601
- CarverHeapPages 属性
 - SQL Anywhere SNMP Extension Agent OID, 972
 - 服务器属性说明, 588
 - 连接属性说明, 566
- CaseSensitive 属性
 - SQL Anywhere SNMP Extension Agent OID, 984
 - 数据库属性说明, 601
- CatalogCollation 属性
 - SQL Anywhere SNMP Extension Agent OID, 984
 - 数据库属性说明, 601
- CBSIZE 连接参数
 - TCP/IP, 136
 - 说明, 253
- CD-ROM
 - 部署, 204
- Certicom
 - 加密客户/服务器通信, 171
- certificate_company 协议选项
 - 说明, 287
- certificate_name 协议选项
 - 说明, 288
- certificate_unit 协议选项
 - 说明, 289
- chained 选项
 - ASE 兼容性, 473
 - Open Client, 1067
 - SQL Anywhere SNMP Extension Agent OID, 987
 - Transact-SQL 兼容性, 473
 - 说明, 488
 - 连接属性说明, 566
- char_charset 别名
 - 关于, 404
- CharSet 连接参数
 - 说明, 252
- CharSet 属性
 - SQL Anywhere SNMP Extension Agent OID, 975, 984
 - 数据库属性说明, 601
 - 服务器属性说明, 588
 - 连接属性说明, 566
- CHAR 归类
 - 关于, 394
- CHAR 数据类型
 - 和主机变量, 480
 - 新数据库的归类序列, 729
 - 新数据库的编码, 729
- checkpoint_time 选项
 - SQL Anywhere SNMP Extension Agent OID, 987
 - 使用, 853
 - 说明, 488
 - 连接属性说明, 566
- CheckpointLogBitmapPagesWritten 属性
 - SQL Anywhere SNMP Extension Agent OID, 980
 - 数据库属性说明, 601
- CheckpointLogBitmapSize 属性
 - SQL Anywhere SNMP Extension Agent OID, 980
 - 数据库属性说明, 601
- CheckpointLogCommitToDisk 属性
 - SQL Anywhere SNMP Extension Agent OID, 980
 - 数据库属性说明, 601
- CheckpointLogPageInUse 属性
 - SQL Anywhere SNMP Extension Agent OID, 980
- CheckpointLogPagesInUse 属性
 - 数据库属性说明, 601
- CheckpointLogPagesRelocated 属性
 - SQL Anywhere SNMP Extension Agent OID, 980

- 数据库属性说明, 601
- CheckpointLogPagesWritten 属性
 - SQL Anywhere SNMP Extension Agent OID, 980
 - 数据库属性说明, 601
- CheckpointLogSavePreimage 属性
 - SQL Anywhere SNMP Extension Agent OID, 980
 - 数据库属性说明, 601
- CheckpointLogSize 属性
 - SQL Anywhere SNMP Extension Agent OID, 980
 - 数据库属性说明, 601
- CheckpointLogWrites 属性
 - SQL Anywhere SNMP Extension Agent OID, 980
 - 数据库属性说明, 601
- CheckpointUrgency 属性
 - SQL Anywhere SNMP Extension Agent OID, 980
 - 数据库属性说明, 601
- Checksum 属性
 - SQL Anywhere SNMP Extension Agent OID, 984
 - 数据库属性说明, 601
- CHECK 约束
 - 卸载数据库, 805
 - 术语定义, 1103
- ChkptFlush 属性
 - SQL Anywhere SNMP Extension Agent OID, 980
 - 数据库属性说明, 601
- ChkptPage 属性
 - SQL Anywhere SNMP Extension Agent OID, 980
 - 数据库属性说明, 601
- Chkpt 属性
 - SQL Anywhere SNMP Extension Agent OID, 980
 - 数据库属性说明, 601
- 重定向器
 - 术语定义, 1096
- 重建
 - 数据库在 Windows Mobile 上, 326
- 重建实用程序 [rebuild]
 - 关于, 760
 - 语法, 760
 - 退出代码, 760
- 重建数据库
 - Windows Mobile, 326
 - 关于, 760
 - 更改归类, 402
- 重命名
 - 事务日志, 849
- 重新调用
 - Interactive SQL 中的命令, 645
- 创建调度向导
 - 使用, 867
- cis_option 选项
 - SQL Anywhere SNMP Extension Agent OID, 987
 - 获取属性值, 566
 - 说明, 488
- cis_rowset_size 选项
 - SQL Anywhere SNMP Extension Agent OID, 987
 - 说明, 489
 - 连接属性说明, 566
- CleanablePagesAdded 属性
 - 数据库属性说明, 601
- CleanablePagesCleaned 属性
 - 数据库属性说明, 601
- CleanableRowsAdded 属性
 - 数据库属性说明, 601
- CleanableRowsCleaned 属性
 - 数据库属性说明, 601
- ClientLibrary 属性
 - 连接属性说明, 566
- ClientNodeAddress 属性
 - 连接属性说明, 566
- ClientPort 属性
 - 连接属性说明, 566
- ClientPort 协议选项
 - 说明, 290
- ClientStmtCacheHits 属性
 - SQL Anywhere SNMP Extension Agent OID, 972
 - 服务器属性说明, 588
 - 连接属性说明, 566
- ClientStmtCacheMisses 属性
 - SQL Anywhere SNMP Extension Agent OID, 972
 - 服务器属性说明, 588
 - 连接属性说明, 566
- close_on_endtrans 选项
 - Open Client, 1067
 - SQL Anywhere SNMP Extension Agent OID, 987
 - Transact-SQL 兼容性, 473
 - 说明, 489
 - 连接属性说明, 566
- Collation 属性
 - SQL Anywhere SNMP Extension Agent OID, 984
 - 数据库属性说明, 601
- collect_statistics_on_dml_updates 选项
 - SQL Anywhere SNMP Extension Agent OID, 987
 - 说明, 490
 - 连接属性说明, 566

-
- CollectStatistics 属性
 - SQL Anywhere SNMP Extension Agent OID, 975
 - 服务器属性说明, 588
 - command_delimiter 选项
 - Interactive SQL 设置, 667
 - 说明, 670
 - CommandLine 属性
 - SQL Anywhere SNMP Extension Agent OID, 975
 - 服务器属性说明, 588
 - CommBufferSize 连接参数
 - TCP/IP, 136
 - 说明, 253
 - commit_on_exit 选项
 - Interactive SQL 设置, 667
 - 说明, 671
 - CommitFile 属性
 - SQL Anywhere SNMP Extension Agent OID, 980
 - 数据库属性说明, 601
 - Commit 属性
 - 连接属性说明, 566
 - COMMIT 语句
 - auto_commit 选项, 668
 - LTM, 1080
 - CommLinks 连接参数
 - 圆括号, 386
 - 说明, 254
 - 选项, 51
 - CommLink 属性
 - 连接属性说明, 566
 - CommNetworkLink 属性
 - 连接属性说明, 566
 - CommProtocol 属性
 - 连接属性说明, 566
 - CompactPlatformVer 属性
 - SQL Anywhere SNMP Extension Agent OID, 975
 - 服务器属性说明, 588
 - CompanyName 属性
 - SQL Anywhere SNMP Extension Agent OID, 975
 - 服务器属性说明, 588
 - CompressionThreshold 连接参数
 - 说明, 256
 - Compression 属性
 - 连接属性说明, 566
 - compression 选项
 - SQL Remote 复制选项, 476
 - 说明, 490
 - Compress 连接参数
 - 说明, 255
 - COMPTRH 连接参数
 - 说明, 256
 - COMP 连接参数
 - 说明, 255
 - conn_auditing 选项
 - SQL Anywhere SNMP Extension Agent OID, 987
 - 使用, 1013
 - 说明, 491
 - 连接属性说明, 566
 - ConnCount 属性
 - SQL Anywhere SNMP Extension Agent OID, 980
 - 数据库属性说明, 601
 - ConnectFailed 系统事件
 - login_procedure 选项示例, 514
 - 示例, 870
 - 说明, 868
 - connection_authentication 选项
 - SQL Anywhere SNMP Extension Agent OID, 987
 - 使用, 71
 - 说明, 491
 - 连接属性说明, 566
 - CONNECTION_PROPERTY 函数
 - 按字母顺序排序的连接属性列表, 566
 - 获得选项值, 464
 - ConnectionName 连接参数
 - 说明, 257
 - Connect 系统事件
 - 说明, 868
 - ConnsDisabled 属性
 - SQL Anywhere SNMP Extension Agent OID, 975, 984
 - 数据库属性说明, 601
 - 服务器属性说明, 588
 - ConsoleLogFile 属性
 - SQL Anywhere SNMP Extension Agent OID, 975
 - 服务器属性说明, 588
 - ConsoleLogMaxSize 属性
 - SQL Anywhere SNMP Extension Agent OID, 975
 - 服务器属性说明, 588
 - continue_after_raisererror 选项
 - ASE 兼容性, 473
 - SQL Anywhere SNMP Extension Agent OID, 987
 - Transact-SQL 兼容性, 473
 - 说明, 492
 - 连接属性说明, 566
 - conversion_error 选项
-

- SQL Anywhere SNMP Extension Agent OID, 987
- Transact-SQL 兼容性, 473
- 说明, 493
- 连接属性说明, 566
- CON 连接参数
 - 说明, 257
- cooperative_commit_timeout 选项
 - SQL Anywhere SNMP Extension Agent OID, 987
 - 说明, 493
 - 连接属性说明, 566
- cooperative_commits 选项
 - SQL Anywhere SNMP Extension Agent OID, 987
 - 说明, 494
 - 连接属性说明, 566
- CPORT 协议选项
 - 说明, 290
- CPU
 - gt 服务器选项, 185
 - 使用的数量, 45
- createcert 实用程序
 - 用法, 1036
 - 语法, 705
- CREATE DATABASE 语句
 - 使用, 21
 - 创建 Windows Mobile 数据库, 325
 - 实用程序数据库, 28
 - 文件管理语句权限, 30
 - 权限, 45
- CREATE DBSPACE 语句
 - 使用, 25
- CREATE DECRYPTED DATABASE 语句
 - 使用, 1022
- CREATE ENCRYPTED DATABASE 语句
 - 与 CREATE ENCRYPTED FILE 语句相比较, 1022
 - 使用, 1021
- CREATE ENCRYPTED FILE 语句
 - 与 CREATE ENCRYPTED DATABASE 语句相比较, 1022
 - 为技术支持加密数据库, 1022
 - 为技术支持解密数据库, 1023
- CREATE EVENT 语句
 - Windows Mobile 上的限制, 337
- CREATE EXISTING TABLE 语句
 - Windows Mobile 上不支持, 337
- CREATE EXTERNLOGIN 语句
 - Windows Mobile 上不支持, 337
- CREATE FUNCTION 语句
 - Windows Mobile 上的限制, 337
- createkey 实用程序
 - 语法, 744
- CREATE LOGIN POLICY 语句
 - 创建新登录策略, 417
 - 提高口令安全性, 1007
- CREATE ON 权限
 - 关于, 428
- CREATE SERVER 语句
 - Windows Mobile 上不支持, 337
- CREATE SUBSCRIPTION 语句
 - 复制服务器, 1080
- CREATE TABLE 语句
 - Windows Mobile 上的限制, 337
- CREATE USER 语句
 - 使用, 431
 - 创建用户并为其分配登录策略, 418
 - 新用户, 430
 - 无口令, 447
- CREATE 权限
 - dbspace, 24
- CSFC5KTNAME 环境变量
 - Kerberos, 107
- CS 连接参数
 - 说明, 252
- CurrentCacheSize 属性
 - SQL Anywhere SNMP Extension Agent OID, 972
 - 服务器属性说明, 588
- CurrentLineNumber 属性
 - 连接属性说明, 566
- CurrentProcedure 属性
 - 连接属性说明, 566
- CurrentRedoPos 属性
 - SQL Anywhere SNMP Extension Agent OID, 984
 - 数据库属性说明, 601
- CURRENT USER
 - 环境设置, 373
- CurIO 属性
 - SQL Anywhere SNMP Extension Agent OID, 980
 - 数据库属性说明, 601
- CurrRead 属性
 - SQL Anywhere SNMP Extension Agent OID, 980
 - 数据库属性说明, 601
- CurrWrite 属性
 - SQL Anywhere SNMP Extension Agent OID, 980
 - 数据库属性说明, 601

- CursorOpen 属性
 - 连接属性说明, 566
- Cursor 属性
 - 连接属性说明, 566
- CyberSafe Kerberos 客户端
 - Unix 支持, 107
 - Windows 支持, 107
- 参考数据库
 - 术语定义, 1096
- 参照完整性
 - 术语定义, 1096
- 操作员
 - 监控器用户, 942
- 策略
 - SELinux 支持, 1004
 - SQL Anywhere 登录, 416
 - 术语定义, 1096
- 插件
 - SQL Anywhere, 630
 - 注册表设置, 374
- 插件模块
 - 术语定义, 1096
- 插入
 - 在 Interactive SQL 中向表中插入行, 655
- 查看
 - TLS 证书, 708
- 查看结果时排除意外符号问题
 - 关于, 387
- 查询
 - Interactive SQL, 641
 - 从 Interactive SQL 打印, 653
 - 优化程序跳过, 530
 - 术语定义, 1096
- 查询编辑器
 - 不支持 Transact-SQL, 651
 - 不支持联合, 651
 - 使用, 650
 - 关于, 649
 - 启动, 650
 - 限制, 651
- 查询内并行机制
 - max_query_tasks 选项, 519
 - 受 -gn 选项影响, 182
- 查询优化
 - 优化程序跳过, 530
 - 捕获最近计划, 230
- 查找
 - Interactive SQL 中的表、列和过程, 648
 - 查找详细信息并请求技术协助
 - 技术支持, xvii
 - 长服务器名
 - 使用 dblocate 查看, 763
 - 超时
 - 故障排除, 145
 - 撤消
 - REMOTE 权限, 439
 - 权限, 439
 - 特权, 439
 - 组成员资格, 445
 - 撤消 Kerberos 登录权限
 - 关于, 111
 - 撤消集成登录权限
 - 关于, 101
 - 撤消日志
 - 关于, 855
 - 成员资格
 - 撤消组成员资格, 445
 - 冲突解决
 - 术语定义, 1096
 - 抽取
 - 术语定义, 1096
 - 抽取实用程序
 - Windows Mobile 上不支持, 340
 - 初始化
 - 数据库, 20
 - 初始化 UI (类型 4) 失败
 - 在 Linux 上显示 SQL Anywhere UI, 219
 - 初始化实用程序 [dbinit]
 - 使用, 22
 - 创建 Windows Mobile 数据库, 324
 - 语法, 729
 - 退出代码, 739
 - 初始化数据库
 - Sybase Central, 20
 - 初始化文件 (INI 文件)
 - 触发器
 - 创建权限, 427
 - 和复制, 504
 - 复制, 503
 - 术语定义, 1097
 - 权限, 438
 - 生成数据库文档, 635
 - 禁用, 180
 - 触发条件

- 定义, 868
- 处理器
 - 使用的数量, 185
 - 多个, 45
 - 并发, 186, 188
- 处理器授权
 - 关于, 766
- 处理事件
 - 关于, 864
- 传输规则
 - 术语定义, 1097
- 传送层安全
 - SSL 支持的版本, 1046
 - 关于, 1031
 - 支持的平台, 1033
 - 效率, 1032
 - 简介, 1032
 - 设置, 1035
- 窗口 (OLAP)
 - 术语定义, 1097
- 创建
 - dbspace, 24
 - Kerberos 登录, 111
 - ODBC 数据源, 使用 [连接] 窗口, 91
 - ODBC 数据源, 使用 dbdsn, 710
 - ODBC 数据源, 使用 ODBC 管理器, 91
 - Windows Mobile 数据库, 322
 - 从 SQL 创建数据库, 21
 - 从命令行创建数据库, 22
 - 使用现有数据库的采用加密表的数据库, 1027
 - 使用现有数据库的高度加密的数据库, 1020
 - 复制服务器主站点连接, 1078
 - 复制服务器复制站点连接, 1078
 - 数据库, 使用 dbinit, 729
 - 新证书, 705
 - 用于复制服务器的复制定义, 1079
 - 用于复制服务器的预订, 1080
 - 用户, 430
 - 登录策略, 417
 - 组, 444
 - 连接配置文件, 88
 - 高度加密的数据库, 1020
- 创建 dbspace 向导
 - 使用, 25
- 创建 Kerberos 登录映射
 - 关于, 111
- 创建 Windows Mobile 数据库
 - CREATE DATABASE 语句, 325
 - dbinit 实用程序, 324
 - Interactive SQL, 325
 - Sybase Central, 323
- 创建备份映像向导
 - 使用, 828
- 创建登录策略
 - 关于, 417
- 创建登录策略向导
 - 使用, 417
- 创建服务向导
 - Windows Mobile 上不支持, 339
 - 使用, 60
- 创建复制定义语句
 - 为复制服务器限定表的所有者, 1079
- 创建活动备份
 - 关于, 821
- 创建集成登录向导
 - 使用, 100
- 创建连接语句
 - 关于, 1078
 - 复制服务器, 1078
- 创建企业根证书
 - 传送层安全, 1039
- 创建签名证书
 - 传送层安全, 1039
- 创建事件向导
 - 使用, 875
- 创建数据库
 - Windows Mobile, 322
 - 关于, 20
 - 创建数据库向导, 20
 - 安全, 1018
 - 选项, 729
- 创建数据库向导
 - Windows Mobile, 323
 - Windows Mobile 上不支持, 339
 - 使用, 20
 - 归类序列列表, 729
- 创建数字证书
 - 传送层安全, 1036
- 创建维护计划向导
 - Windows Mobile 上有限支持, 339
 - 使用, 845
- 创建用户
 - 关于, 418
- 创建用户向导

- 使用, 431
- 创建者 ID
 - 术语定义, 1097
- 创建组向导
 - 使用, 444
- 磁带驱动器
 - 备份数据库, 826
- 磁盘
 - 从故障中恢复, 833
 - 碎片整理和性能, 26
- 磁盘高速缓存
 - 操作系统, 215
- 磁盘镜像
 - 事务日志, 15
- 磁盘空间
 - 文件系统已满回调函数, 175
 - 事件示例, 870
 - 使用 dbinfo 确定磁盘上表的大小, 728
- 磁盘控制器
 - 事务日志管理, 15
- 磁盘损坏
 - 关于, 846
- 磁盘已满
 - 回调函数, 175
 - 写入事务日志时出错, 15
- 从不写入模式
 - 数据库服务器, 188
- 从登录策略中删除用户
 - 关于, 418
- 从主服务器故障恢复
 - 关于, 900
- 存储过程
 - 使用 SQL Anywhere SNMP Extension Agent 执行, 965
 - 安全功能, 1004
 - 术语定义, 1097
 - 生成数据库文档, 635
 - 设置权限, 437
- 存储卡
 - Windows Mobile, 310
- 存储区域网络
 - 存储数据库文件, 14
- 错误
 - Interactive SQL, 647
 - Transact-SQL 过程, 528
 - 事件处理程序行为, 874
 - 在 Interactive SQL 中, 679

- 将报告提交给 iAnywhere, 76
- 提供反馈, xvi
- 错误报告
 - 关于, 76
- 错误处理
 - Interactive SQL, 679
 - Transact-SQL 过程, 528

D

- DAC
 - 嵌套视图和表的规则, 454
- database_authentication 选项
 - SQL Anywhere SNMP Extension Agent OID, 987
 - 使用, 71
 - 说明, 495
 - 连接属性说明, 566
- DatabaseCleaner 属性
 - SQL Anywhere SNMP Extension Agent OID, 984
 - 数据库属性说明, 601
- DatabaseFile 连接参数
 - 嵌入式数据库, 118
 - 说明, 258
- DatabaseKey 连接参数
 - 说明, 259
- DatabaseName 连接参数
 - 说明, 260
- DatabaseName 协议选项
 - 说明, 291
- DatabaseStart 系统事件
 - 说明, 868
- DatabaseSwitches 连接参数
 - 说明, 261
- DataSourceName 连接参数
 - Windows Mobile, 94
 - 说明, 262
- date_format 选项
 - Open Client, 1067
 - SQL Anywhere SNMP Extension Agent OID, 987
 - Transact-SQL 兼容性, 473
 - 说明, 496
 - 连接属性说明, 566
- date_order 选项
 - Open Client, 1067
 - SQL Anywhere SNMP Extension Agent OID, 987
 - Transact-SQL 兼容性, 473
 - 说明, 497
 - 连接属性说明, 566

- daylight savings time
 - 调度事件, 873
- db_charset
 - 关于, 386
- DB_PROPERTY 函数
 - 按字母顺序排序的数据库属性列表, 601
- DBA 权限
 - 为新数据库指定 DBA 用户, 729
 - 安全提示, 1005
 - 术语定义, 1098
- DBA 特权
 - 不可继承, 425
 - 关于, 425
 - 授予, 433
- dbbackup 实用程序
 - 完全备份, 820
 - 客户端备份, 830
 - 接收错误, 698
 - 活动备份, 821
 - 语法, 698
 - 退出代码, 702
- dbcc 函数
 - 使用, 792
- dbconsole 实用程序
 - Mac OS X 硬件要求, 688
 - 使用, 687
 - 启动, 687
 - 语法, 780
 - 软件更新, 690
- dbctrs11.dll
 - 在 Vista 上可能需要权限提升, 39
- DBDiskSpace 系统事件
 - 示例, 870
 - 说明, 868
- dbdsn 实用程序
 - 使用, 92
 - 在 Vista 上可能需要权限提升, 38
 - 系统信息文件, 713
 - 语法, 710
 - 退出代码, 713
- dbelevat11.exe
 - 在 Vista 上可能需要权限提升, 38
 - 必须包括在 Vista 的部署中, 39
- dbeng11
 - 个人数据库服务器, 36
 - 命令行, 148
 - 授权, 766
 - 语法, 148
- dberase 实用程序
 - 使用, 32
 - 语法, 722
 - 退出代码, 723
- dbfhide 实用程序
 - 语法, 724
- DBFileFragments 属性
 - SQL Anywhere SNMP Extension Agent OID, 984
 - 数据库属性说明, 601
- DBF 连接参数
 - 嵌入式数据库, 118
 - 说明, 258
- dbhist 实用程序
 - 语法, 726
 - 退出代码, 727
- dbicu11.dll
 - 创建 Windows Mobile 数据库, 323
 - 卸载 Windows Mobile 上的数据库, 327
 - 避免 ICU 在 Windows Mobile 上, 324
- dbicudt11.dll
 - 创建 Windows Mobile 数据库, 323
 - 卸载 Windows Mobile 上的数据库, 327
 - 避免 ICU 在 Windows Mobile 上, 324
- dbinfo
 - 用于确定磁盘上表的大小, 728
- dbinfo 实用程序
 - 语法, 728
 - 退出代码, 728
- dbinit 实用程序
 - 使用, 22
 - 创建 Windows Mobile 数据库, 324
 - 语法, 729
 - 退出代码, 739
- dbisql.com
 - 关于, 742
- dbisql.exe
 - 关于, 742
 - 快速启动程序选项, 686
- dbisqlc 实用程序
 - 支持的平台, 720
 - 语法, 720
- dbisql 实用程序
 - (参见 Interactive SQL)
 - (参见 Interactive SQL 实用程序 [dbisql])
 - 关于, 636
 - 支持的平台, 742

语法, 740
退出代码, 742

DBKEY 连接参数
说明, 259

dblang 实用程序
关于, 745
在启用快速启动程序时使用, 746
语法, 745
退出代码, 746

DBLauncher
在 Mac OS X 上启动数据库服务器, 117

dblgen11.res
定位, 370

dblic 实用程序
在 Vista 上可能需要权限提升, 38
语法, 766
退出代码, 767

dblocate 实用程序
语法, 763
退出代码, 765

dblog 实用程序
事务日志镜像, 16
命令行, 794
审计, 1017
语法, 792
退出代码, 794

dbltn 实用程序
语法, 748
退出代码, 749

dbmlsync 实用程序
TLS, 1050

DBMS
术语定义, 1111

DBNS
定义, 128

dbns11 实用程序
使用, 128
语法, 703

DBNumber 属性
连接属性说明, 566

DBN 连接参数
说明, 260

DBN 协议选项
说明, 291

dbodbc11.dll
在 Vista 上可能需要权限提升, 39

dboledb11.dll
在 Vista 上可能需要权限提升, 39

dboledb11.dll
在 Vista 上可能需要权限提升, 39

dbo 用户
关于, 448
系统对象和卸载实用程序, 804

dbping_r 实用程序
在 Unix 上使用, 757

dbping 实用程序
使用, 130
语法, 757
退出代码, 759

dbrunsql 实用程序
语法, 761

dbsnmp11.dll
关于, 956

dbspace
CREATE ON 权限, 428
default_dbpace 选项, 499
使用 -ds 服务器选项指定位置, 237
关于, 23
创建, 24
删除, 27
变更, 26
权限, 24
用于大型数据库, 23
限制, 614
预定义, 13

dbspaces
卸载时更改文件名, 795
术语定义, 1098

dbspawn 实用程序
语法, 781
退出代码, 781

dbsrv11
Windows Mobile, 329
传送层安全, 1042
命令行, 148
授权, 766
网络数据库服务器, 36
语法, 148

dbsrv11.nlm
关于, 36

dbstop 实用程序
与 SQLCONNECT 一起使用, 784
使用, 53
权限, 181

- 语法, 783
- 退出代码, 784
- dbsupport.ini 文件
 - 关于, 787
- dbsupport 实用程序
 - SADIAGDIR 环境变量, 353
 - 使用, 76
 - 语法, 785
- dbsvc 实用程序
 - Linux 语法, 769
 - Linux 选项, 769
 - Windows 语法, 773
 - Windows 选项, 773
 - 在 Vista 上可能需要权限提升, 38
 - 退出代码, 777
- DBS 连接参数
 - 说明, 261
- dbtran 实用程序
 - 事务日志, 834
 - 使用, 837
 - 命令行, 755
 - 审计, 1017
 - 未提交的更改, 834
 - 检索审计信息, 1014
 - 语法, 753
 - 退出代码, 756
- dbunload 实用程序
 - dbspace 文件名, 795
 - 语法, 795
 - 退出代码, 805
- dbupgrad 实用程序
 - 语法, 808
 - 退出代码, 809
- dbvalid 实用程序
 - 语法, 810
- dbvalid 实用程序
 - 使用, 820
 - 退出代码, 812
- dbversion 实用程序
 - 语法, 813
- dbvss11.exe
 - SQL Anywhere VSS 写入程序, 220, 829
- dbxtract 实用程序
 - Windows Mobile 上不支持, 340
- DCX
 - 关于, xii
- DDL
 - 术语定义, 1110
- Deadlock 系统事件
 - 说明, 868
- debug_messages 选项
 - SQL Anywhere SNMP Extension Agent OID, 987
 - 说明, 498
 - 连接属性说明, 566
- DebuggingInformation 属性
 - SQL Anywhere SNMP Extension Agent OID, 975
 - 服务器属性说明, 588
- DECRYPT 函数
 - 用于解密列, 1024
- dedicated_task 选项
 - SQL Anywhere SNMP Extension Agent OID, 987
 - 说明, 499
 - 连接属性说明, 566
- default_dbpace 选项
 - SQL Anywhere SNMP Extension Agent OID, 987
 - 指定数据库对象的位置, 23
 - 说明, 499
 - 连接属性说明, 566
- default_isql_encoding 选项
 - Interactive SQL 设置, 667
 - 说明, 671
- default_timestamp_increment 选项
 - SQL Anywhere SNMP Extension Agent OID, 987
 - 用于 MobiLink 同步, 500
 - 说明, 500
 - 连接属性说明, 566
- DefaultCollation 属性
 - SQL Anywhere SNMP Extension Agent OID, 975
 - 关于, 28
 - 服务器属性说明, 588
- DefaultNcharCollation 属性
 - SQL Anywhere SNMP Extension Agent OID, 975
 - 服务器属性说明, 588
- delayed_commit_timeout 选项
 - SQL Anywhere SNMP Extension Agent OID, 987
 - 说明, 501
 - 连接属性说明, 566
- delayed_commits 选项
 - SQL Anywhere SNMP Extension Agent OID, 987
 - 说明, 501
 - 连接属性说明, 566
- delete_old_logs 选项
 - 重置截断偏移, 792
 - SQL Remote 复制和同步选项, 476

事务日志选项, 792
 使用, 1090
 复制代理选项, 477
 说明, 502
DELETE 权限
 关于, 428
 授予, 434
DELETE 语句
 LTM, 1084
 在 Interactive SQL 中生成, 649
Delphi
 BINARY 列, 524
Delphi 连接参数
 ODBC 连接参数说明, 714
demo.db 文件
 运行个人服务器示例, 5
DER 编码的 PKI 对象
 查看, 708
DescribeCursor 连接参数
 ODBC 连接参数说明, 714
Description 连接参数
 ODBC 连接参数说明, 714
调度
 创建调度向导, 867
 为事件定义, 866
 事件, 864
 关于, 866
 内部, 873
 备份, 843, 844
 定义, 864, 866
 服务器关闭, 214
 简介, 864
调度事件
 daylight savings time, 873
 关于, 866
DisableMultiRowFetch 连接参数
 说明, 263
Disconnect 系统事件
 说明, 868
DISCONNECT 语句
 使用, 132
DiskReadHintPages 属性
 数据库属性说明, 601
 连接属性说明, 566
DiskReadHintScatterLimit 属性
 服务器属性说明, 588
DiskReadHint 属性
 数据库属性说明, 601
 连接属性说明, 566
DiskReadIndInt 属性
 SQL Anywhere SNMP Extension Agent OID, 980
 数据库属性说明, 601
 连接属性说明, 566
DiskReadIndLeaf 属性
 SQL Anywhere SNMP Extension Agent OID, 980
 数据库属性说明, 601
 连接属性说明, 566
DiskReadTable 属性
 SQL Anywhere SNMP Extension Agent OID, 980
 数据库属性说明, 601
 连接属性说明, 566
DiskReadWorkTable
 数据库属性说明, 601
DiskReadWorkTable 属性
 SQL Anywhere SNMP Extension Agent OID, 980
 连接属性说明, 566
DiskRead 属性
 SQL Anywhere SNMP Extension Agent OID, 972, 980
 数据库属性说明, 601
 服务器属性说明, 588
 连接属性说明, 566
DiskRetryReadScatter 属性
 数据库属性说明, 601
 服务器属性说明, 588
DiskRetryRead 属性
 服务器属性说明, 588
DiskRetryWrite 属性
 服务器属性说明, 588
DiskSyncRead 属性
 数据库属性说明, 601
 连接属性说明, 566
DiskSyncWrite 属性
 数据库属性说明, 601
 连接属性说明, 566
DiskWaitRead 属性
 数据库属性说明, 601
 连接属性说明, 566
DiskWaitWrite 属性
 数据库属性说明, 601
 连接属性说明, 566
DiskWriteHintPages 属性
 数据库属性说明, 601
 连接属性说明, 566

- DiskWriteHint 属性
 - 数据库属性说明, 601
 - 连接属性说明, 566
- DiskWrite 属性
 - SQL Anywhere SNMP Extension Agent OID, 980
 - 数据库属性说明, 601
 - 连接属性说明, 566
- DLL
 - 位置, 370
- DML
 - 术语定义, 1110
- DMRF 连接参数
 - 说明, 263
- DoBroadcast 协议选项
 - 说明, 291
- DOBROAD 协议选项
 - 说明, 291
- DocCommentXchange (DCX)
 - 关于, xii
- 动态调整高速缓存大小
 - 对数据库服务器禁用, 159
- Driver 连接参数
 - ODBC 连接参数说明, 714
- DriveType 属性
 - SQL Anywhere SNMP Extension Agent OID, 984
 - 数据库属性说明, 601
- DROP CONNECTION 语句
 - 使用, 132
- DROP DATABASE 语句
 - Windows Mobile 上不支持, 337
 - 使用, 32
- DROP LOGIN POLICY 语句
 - 删除登录策略, 420
- DROP SERVER 语句
 - Windows Mobile 上不支持, 337
- DSEdit 实用程序
 - 不随 SQL Anywhere 提供, 1060
 - 使用, 1062
 - 关于, 1060
 - 启动, 1062
 - 条目, 1064
 - 设置 Open Server, 1074
- DSN 连接参数
 - Windows Mobile, 94
 - 关于, 90
 - 说明, 262
- DUMMY
 - 系统表权限, 458
- DYLD_LIBRARY_PATH 环境变量
 - 说明, 346
- 打印
 - Interactive SQL, 653
- 大小写敏感性
 - 国际方面, 388
- 代理
 - 关于, 957
- 代理 ID
 - 术语定义, 1097
- 代理表
 - 术语定义, 1097
- 代码编辑器
 - 关于, 625
 - 字体, 626
 - 打开, 626
 - 自定义外观, 626
 - 键盘快捷方式, 626
- 代码页
 - ANSI, 386
 - default_isql_encoding 选项, 671
 - Interactive SQL [dbisql] 实用程序, 740
 - OEM, 386
 - Windows, 386
 - 定义, 384
 - 建议在 Unix 平台上使用, 409
 - 建议在 Windows 平台上使用, 408
 - 术语定义, 1097
 - 概述, 385
- 单步
 - 关于, 642
- 单步执行
 - Interactive SQL, 641
- 单字节字符集
 - 关于, 385
- 档案
 - (参见 备份)
 - 直接备份到磁带上, 828
- 档案备份
 - 关于, 823
 - 定义的, 823
 - 恢复, 835
- 导出
 - 连接配置文件, 89
- 导出数据
 - 输出格式, 680

- 导入
 - 连接配置文件, 89
 - 登录
 - Kerberos, 107
 - 集成, 99
 - 登录策略
 - 从根策略继承, 416
 - 修改, 419
 - 关于, 416
 - 分配给现有用户, 418
 - 创建, 417
 - 创建新用户时分配, 418
 - 删除, 418, 420
 - 只读数据库, 420
 - 根登录策略, 416
 - 根策略中的替代策略选项, 416
 - 登录映射
 - Kerberos, 107
 - 集成登录, 99
 - 第一行优化选项
 - optimization_goal, 528
 - 递归查询
 - max_recursive_iterations 选项, 520
 - 定制归类
 - ICU, 393
 - Windows Mobile 上的有限支持, 321
 - 动态 SQL
 - 术语定义, 1098
 - 动态陷阱
 - 关于, 966
 - 逗号分隔的文件
 - input_format 选项, 672
 - output_format 选项, 680
 - 读取
 - TLS 证书, 708
 - 度量
 - 监控器, 928
 - 编辑收集间隔, 936
 - 端对端 (见 端对端加密)
 - 端对端加密
 - createkey 实用程序, 744
 - MobiLink 关于, 1047
 - 端口参数
 - LDAP, 139
 - 端口号
 - ServerPort [PORT] 协议选项, 303
 - TCP/IP, 221
 - 将 SQL Anywhere 用作 Open Server 时的 TCP/IP, 1061
 - 数据库服务器, 303
 - 断开连接
 - 从数据库, 132
 - 停止数据库, 55
 - 其他用户与数据库的连接, 132
 - 用户通过 SQL Anywhere 控制台实用程序, 780
 - 断开与数据库的连接
 - 关于, 132
 - 堆栈大小
 - 外部函数, 180
 - 最大值, 185
 - 堆栈溢出
 - 错误, 185
 - 对象
 - 限定名, 450
 - 对象标识符 (见 OID)
 - 对象树
 - 术语定义, 1098
 - 多处理
 - 控制线程, 47
 - 多处理器支持
 - 控制线程, 47
 - 服务器选项, 44
 - 多个数据库
 - DSEdit 条目, 1064
 - 多路广播地址
 - IPv6 支持, 135
 - 多任务
 - 控制线程, 47
 - 多字节字符集
 - 使用, 392
 - 关于, 385
- ## E
- EBF
 - 术语定义, 1098
 - ECC
 - 支持, 1033
 - ECC 选项
 - dbeng11 -ec, 172
 - dsrv11 -ec, 172
 - ECC 证书
 - 创建, 705
 - 查看, 708
 - echo 选项

- Interactive SQL 设置, 667
 - 说明, 672
- Elevate 连接参数
 - 说明, 264
- EncryptedPassword 连接参数
 - 说明, 264
- EncryptionScope 属性
 - SQL Anywhere SNMP Extension Agent OID, 984
 - 数据库属性说明, 601
- Encryption 连接参数
 - 保护客户端/服务器通信安全, 1043
 - 说明, 265
- Encryption 属性
 - SQL Anywhere SNMP Extension Agent OID, 984
 - 说明, 601
 - 连接属性说明, 566
- ENCRYPT 函数
 - 用于加密列, 1024
- ENC 连接参数
 - 保护客户端/服务器通信安全, 1043
 - 说明, 265
- EngineName 连接参数
 - (参见 ServerName 连接参数)
 - 说明, 280
 - 连接到镜像数据库, 896
- ENG 连接参数
 - (参见 ServerName 连接参数)
 - 嵌入式数据库, 119
 - 说明, 280
 - 连接到镜像数据库, 896
- ENP 连接参数
 - 说明, 264
- ERRORLEVEL 环境变量
 - Interactive SQL 返回代码, 740
- ER 图选项卡
 - 关于, 632
- escape_character 选项
 - ASE 兼容性, 473
 - Open Client, 1067
 - SQL Anywhere SNMP Extension Agent OID, 987
 - Transact-SQL 兼容性, 473
 - 说明, 502
 - 连接属性说明, 566
- ESQL 示例
 - 使用, 316
- EventName 属性
 - 连接属性说明, 566
- EventTypeDesc
 - 服务器属性说明, 588
- EventTypeDesc 属性
 - SQL Anywhere SNMP Extension Agent OID, 975
- EventTypeName
 - 服务器属性说明, 588
- EventTypeName 属性
 - SQL Anywhere SNMP Extension Agent OID, 975
- ExchangeTasksCompleted 属性
 - 服务器属性说明, 588
- ExchangeTasks 属性
 - SQL Anywhere SNMP Extension Agent OID, 972
 - 服务器属性说明, 588
- exclude_operators 选项
 - SQL Anywhere SNMP Extension Agent OID, 987
 - 说明, 503
 - 连接属性说明, 566
- ExprCacheAbandons 属性
 - 数据库属性说明, 601
 - 连接属性说明, 566
- ExprCacheDropsToReadOnly 属性
 - 数据库属性说明, 601
 - 连接属性说明, 566
- ExprCacheEvicts 属性
 - 数据库属性说明, 601
 - 连接属性说明, 566
- ExprCacheHits 属性
 - 数据库属性说明, 601
 - 连接属性说明, 566
- ExprCacheInserts 属性
 - 数据库属性说明, 601
 - 连接属性说明, 566
- ExprCacheLookups 属性
 - 数据库属性说明, 601
 - 连接属性说明, 566
- ExprCacheResumesOfReadWrite 属性
 - 数据库属性说明, 601
 - 连接属性说明, 566
- ExprCacheStarts 属性
 - 数据库属性说明, 601
 - 连接属性说明, 566
- ExtendDB 属性
 - SQL Anywhere SNMP Extension Agent OID, 980
 - 数据库属性说明, 601
- extended_join_syntax 选项
 - SQL Anywhere SNMP Extension Agent OID, 987
 - 说明, 503

- 连接属性说明, 566
- ExtendTempWrite 属性
 - SQL Anywhere SNMP Extension Agent OID, 980
 - 数据库属性说明, 601
- external_remote_options 选项
 - SQL Remote 选项, 503
- F**
- 分布式事务协调器
 - 恢复无, 213
- FILE
 - 术语定义, 1099
- FileDataSourceName 连接参数
 - Windows Mobile, 94
 - 引用文件数据源, 90
 - 说明, 267
- FILEDSN 连接参数
 - Windows Mobile, 94
 - 引用文件数据源, 90
 - 说明, 267
- FileSize 属性
 - SQL Anywhere SNMP Extension Agent OID, 984
 - 数据库属性说明, 601
- File 属性
 - SQL Anywhere SNMP Extension Agent OID, 984
 - 数据库属性说明, 601
- FILE 消息类型
 - 术语定义, 1099
- Finder
 - 设置环境变量, 344
- FIPS
 - dbeng11 -ec, 171
 - dbeng11 -fips, 176
 - dbinit -ea, 729
 - dbsrv11 -ec, 171
 - dbsrv11 -fips, 176
 - SQL_FLAGGER_ERROR 选项, 545
 - Web 服务, 224
 - 关于, 1033
 - 对数据库文件进行加密, 795
 - 支持, 1033
- FIPS 140-2 证书
 - 关于, 1033
- FipsMode 属性
 - SQL Anywhere SNMP Extension Agent OID, 975
 - 服务器属性说明, 588
- FIPS 协议选项
 - dbeng11 -ec, 171
 - dbsrv11 -ec, 171
- FIPS 选项
 - AES256_FIPS 加密算法, 176
 - AES_FIPS 加密算法, 176
 - 数据库服务器, 176
- fire_triggers 选项
 - SQL Anywhere SNMP Extension Agent OID, 987
 - Transact-SQL 兼容性, 473
 - 说明, 504
 - 连接属性说明, 566
- first_day_of_week 选项
 - SQL Anywhere SNMP Extension Agent OID, 987
 - 说明, 504
 - 连接属性说明, 566
- FirstOption 属性
 - SQL Anywhere SNMP Extension Agent OID, 975
 - 服务器属性说明, 588
- FIXED 文件格式
 - input_format 选项, 672
 - Interactive SQL 输出, 680
- Flagger (*见* SQL Flagger)
- for_xml_null_treatment 选项
 - SQL Anywhere SNMP Extension Agent OID, 987
 - 说明, 505
 - 连接属性说明, 566
- force_view_creation 选项
 - SQL Anywhere SNMP Extension Agent OID, 987
 - 说明, 506
 - 连接属性说明, 566
- ForceStart 连接参数
 - 说明, 268
- FORCE 连接参数
 - 说明, 268
- FreeBuffers 属性
 - SQL Anywhere SNMP Extension Agent OID, 972
 - 服务器属性说明, 588
- FreePages 属性
 - SQL Anywhere SNMP Extension Agent OID, 984
 - 数据库属性说明, 601
- FullCompare 属性
 - SQL Anywhere SNMP Extension Agent OID, 980
 - 数据库属性说明, 601
 - 连接属性说明, 566
- FunctionMaxParms 属性
 - 服务器属性说明, 588
- FunctionMinParms 属性

- 服务器属性说明, 588
- FunctionName 属性
 - 服务器属性说明, 588
- 发布
 - 术语定义, 1098
- 发布更新
 - 术语定义, 1098
- 发布者
 - 术语定义, 1099
- 发起文件
 - Unix, 344
- 发送
 - 监控器警告电子邮件, 947
- 发送电子邮件
 - 监控器用户, 943
 - 监控器警告通知, 947
- 法定数目
 - 数据库镜像, 881
- 翻译日志文件实用程序 [dbtran]
 - 使用, 837
 - 审计, 1017
 - 恢复未提交的操作, 834
 - 检索审计信息, 1014
 - 语法, 753
 - 退出代码, 756
- 翻译日志文件向导
 - Windows Mobile 上不支持, 339
 - 使用, 834
- 反 SQL 日志 (// 回退日志)
- 反馈
 - 报告错误, xvi
 - 提供, xvi
 - 文档, xvi
 - 请求更新, xvi
- 返回代码
 - 重建 [rebuild] 实用程序, 760
 - Interactive SQL [dbisql] 实用程序, 742
 - 校验实用程序 dbvalid (dbvalid), 812
 - ping [dbping] 实用程序, 759
 - Windows 服务 [dbsvc] 实用程序, 777
 - 事务日志 [dblog] 实用程序, 794
 - 信息 [dbinfo] 实用程序, 728
 - 停止服务器 [dbstop] 实用程序, 784
 - 初始化 [dbinit] 实用程序, 739
 - 升级 [dbupgrad] 实用程序, 809
 - 卸载 [dbunload] 实用程序, 805
 - 后台启动服务器 [dbspawn] 实用程序, 781
 - 备份 [dbbackup] 实用程序, 702
 - 数据源 [dbdsn] 实用程序, 713
 - 日志传送管理器 [dbltn] 实用程序, 749
 - 服务器授权 [dblic] 实用程序, 767
 - 服务器枚举 [dblocate] 实用程序, 765
 - 消除 [dberase] 实用程序, 723
 - 直方图 [dbhist] 实用程序, 727
 - 翻译日志文件 [dbtran] 实用程序, 756
 - 语言 [dblang] 实用程序, 746
- 范围
 - (参见 限制)
- 防火墙
 - BroadcastListener [BLISTENER] 协议选项, 286
 - ClientPort [CPORT] 协议选项, 290
 - HOST [IP] 协议选项, 292
 - LDAP 协议选项, 295
 - ServerPort [PORT] 协议选项, 303
 - 连接, 136, 765
 - 连接穿过, 138
- 访问计划
 - 控制优化程序的使用, 529
- 访问数据库
 - 安全功能, 1004
- 非链接模式
 - chained 选项, 488
- 非英语数据库
 - 创建, 401
- 分布式事务
 - 征用超时, 214
 - 恢复, 213
- 分号
 - 在连接字符串内使用, 81
- 分配
 - 登录策略在创建新用户时, 418
- 分配给个别用户的登录策略
 - 卸载 [dbunload] 实用程序, 804
- 分析
 - 执行权限, 425
- 分析树
 - 术语定义, 1099
- 封锁
 - 关于, 939
- 符号
 - 查看结果时排除意外符号问题, 387
- 服务
 - Linux 上的设置类型, 770
 - Linux 列出, 769

- Linux 启动, 769
- Linux 设置依赖性, 770
- Windows, 59, 773
- 为 Windows 添加, 60
- 事件日志, 42
- 从 Sybase Central 创建, 60
- 使用 Windows 服务管理器, 65
- 依赖性, 66, 67
- 停止, 65
- 关于, 57
- 列出, 773
- 删除, 61
- 参数, 61
- 可执行文件, 64
- 启动, 65, 773
- 启动失败, 62
- 启动选项, 62
- 启动顺序, 67
- 在 Vista 上运行, 39
- 在 Windows 上设置类型, 774
- 多个, 66
- 安全, 63
- 帐户, 63
- 数据库服务器, 57
- 服务 [dbsvc] 实用程序, 773
- 术语定义, 1099
- 桌面上的图标, 63
- 注册表设置, 373
- 添加新数据库, 64
- 符合条件的程序, 59
- 管理, 60
- 组, 66
- 设置依赖性, 774
- 设置组依赖性, 774
- 轮询, 65
- 选项, 62
- 配置, 61
- 服务创建实用程序 (见 服务实用程序 [dbsvc])
- 服务监控器 (见 SQL Anywhere 监控器)
- 服务器
 - (参见 数据库服务器)
 - 从批处理文件启动, 781
 - 停止数据库, 55
 - 只读访问, 数据库镜像, 241
 - 名称截断长度, 196
 - 名称限制, 196
 - 启动支持传送层安全, 1042
 - 启动未连接的数据库, 54
 - 定位, 126, 763
 - 属性, 588
 - 指定替代名, 242
 - 禁用数据库功能, 1010
 - 管理, 773
 - 自动启动, 126
 - 通过数据库镜像提供高可用性, 880
 - 限制连接, 182
- 服务器参数
 - LDAP, 139
- 服务器地址
 - DSEdit, 1064
- 服务器定位实用程序 (见 服务器枚举实用程序 [dblocate])
- 服务器端
 - ec 服务器选项, 171
 - ek 服务器选项, 238
 - ep 服务器选项, 173
 - es 服务器选项, 174
 - 备份, 825
 - 备份快速入门, 818
 - 并行备份, 855
- 服务器管理请求
 - 术语定义, 1099
- 服务器监控
 - 关于, 913
- 服务器枚举实用程序 [dblocate]
 - 语法, 763
 - 退出代码, 765
- 服务器名
 - n 选项, 195
 - 停止 [dbstop] 实用程序, 783
 - 区分大小写, 44
 - 服务器枚举 [dblocate] 实用程序, 763
- 服务器启动的同步
 - 术语定义, 1099
- 服务器启动选项窗口
 - 在 Windows Mobile 上使用, 329
- 服务器授权实用程序 [dblic]
 - 在 Vista 上可能需要权限提升, 38
 - 语法, 766
 - 退出代码, 767
- 服务器属性
 - SQL Anywhere SNMP Extension Agent 的 OID, 975

- 使用 SQL Anywhere SNMP Extension Agent 检索, 964
- 使用 SQL Anywhere SNMP Extension Agent 设置, 965
- 区分大小写, 588
- 报告, 757
- 按字母顺序排序的列表, 588
- 服务器统计信息
 - SQL Anywhere SNMP Extension Agent 的 OID, 972
 - 使用 SQL Anywhere SNMP Extension Agent 检索, 964
- 服务器消息
 - 重命名和重新启动日志文件, 198
 - 在 Linux 上显示, 215, 217
 - 在 Linux 上查看, 215, 217, 219
 - 在 Mac OS X 上显示, 218
 - 在 Mac OS X 上查看, 218
 - 在 Solaris 上显示, 215
 - 在 Solaris 上查看, 215
 - 显示, 203
 - 记录启动错误, 197
 - 输出到文件, 196, 199
 - 限制日志文件的大小, 198
 - 高速缓存预热, 166
- 服务器消息存储库
 - 术语定义, 1099
- 服务器消息和已执行的 SQL 窗格
 - 关于, 42
- 服务器消息日志
 - 针对数据库服务器进行配置, 40
- 服务器信息
 - sasrv.ini, 129
- 服务器选项
 - Windows Mobile 上不支持的选项, 338
 - 为 Windows Mobile 数据库指定, 1029
 - 常用选项, 43
 - 恢复, 175
 - 数据库, 234
- 服务器验证
 - MobiLink 传送层安全, 1049
 - SQL Anywhere 传送层安全, 1043
- 服务器证书
 - 在传送层安全中使用全局证书, 1040
- 服务实用程序 [dbsvc]
 - Linux 语法, 769
 - Linux 选项, 769
- Windows 语法, 773
- Windows 选项, 773
 - 在 Vista 上可能需要权限提升, 38
 - 退出代码, 777
- 服务组
 - 关于, 66
- 复制
 - 重建加密的数据库, 807
 - dbcc, 794
 - Interactive SQL 中的行, 656
 - SQL Anywhere 插件中的数据库对象, 631
 - SQL Remote 选项, 476
 - 为主站点启用, 1076
 - 为复制服务器创建复制定义, 1079
 - 事务日志管理, 847
 - 关于复制服务器, 1070
 - 备份过程, 847, 1090
 - 复制服务器, 794
 - 复制服务器事务日志管理, 1090
 - 复制服务器备份过程, 1090
 - 复制服务器存储过程, 1086
 - 复制服务器的整个数据库, 1091
 - 复制服务器缓冲, 1087
 - 复制服务器过程, 1085, 1086
 - 定义, 1084
 - 数据库到 Windows Mobile 设备, 325
 - 日志传送管理器, 748
 - 术语定义, 1099
 - 触发器操作, 503, 504
- 复制代理
 - 备份, 847
 - 数据库选项, 477
 - 日志传送管理器实用程序 [dbltm] 语法, 748
 - 术语定义, 1099
 - 标识符, 1083
- 复制单元格
 - 在 Interactive SQL 中, 656
- 复制服务器
 - rssetup.sql 脚本, 1075
 - SQL Anywhere 字符集, 1083
 - SQL Anywhere 归类, 1083
 - SQL Anywhere 配置, 1081
 - 主站点, 1071, 1078
 - 事务日志管理, 1090
 - 关于, 1069, 1070
 - 准备 SQL Anywhere 数据库, 1077
 - 创建复制定义, 1079

- 创建连接, 1078
- 创建预订, 1080
- 受支持的版本, 1071
- 启动 SQL Anywhere 服务器, 1074
- 备份过程, 1090
- 复制整个数据库, 1091
- 复制站点, 1071, 1078
- 复制过程, 1085, 1086
- 支持, 1058
- 术语定义, 1100
- 特点, 1070
- 设置 SQL Anywhere 数据库, 1081
- 复制服务器的代理 (见 LTM)
- 复制列
 - 在 Interactive SQL 中, 656
- 复制频率
 - 术语定义, 1100
- 复制数据库
 - 到 Windows Mobile 设备, 325
 - 安全问题, 115
- 复制消息
 - 术语定义, 1100
- 复制选定行
 - 在 Interactive SQL 中, 656
- 复制选项
 - replicate_all, 539
 - SQL Remote blob_threshold, 486
 - SQL Remote compression, 490
 - SQL Remote delete_old_logs, 502
 - SQL Remote external_remote_options, 503
 - SQL Remote qualify_owners, 536
 - SQL Remote quote_all_identifiers, 537
 - SQL Remote replication_error, 539
 - SQL Remote replication_error_piece, 540
 - SQL Remote save_remote_passwords, 543
 - SQL Remote sr_date_format, 547
 - SQL Remote sr_time_format, 548
 - SQL Remote sr_timestamp_format, 549
 - SQL Remote subscribe_by_remote, 550
 - SQL Remote verify_all_columns, 560
 - SQL Remote verify_threshold, 563
 - SQL Remote 列表, 476
 - 分类, 466
 - 初始设置, 465
 - 复制代理 delete_old_logs, 502
- 复制站点
 - 使用 LTM, 1072

- 创建, 1073
- 复制服务器, 1071
- 添加复制服务器信息, 1077
- 复制最少列
 - 支持, 1085
- 负权限
 - 不支持, 424

G

- GetData 属性
 - SQL Anywhere SNMP Extension Agent OID, 980
 - 数据库属性说明, 601
 - 连接属性说明, 566
- GetTypeInfoChar 连接参数
 - ODBC 连接参数说明, 714
- global_database_id 选项
 - SQL Anywhere SNMP Extension Agent OID, 987
 - 说明, 506
 - 连接属性说明, 566
- GlobalAutoIncrement 系统事件
 - 说明, 868
- GlobalDBID 属性
 - SQL Anywhere SNMP Extension Agent OID, 984
 - 数据库属性说明, 601
- go
 - Interactive SQL 分隔符, 670
 - 用法, 643
- GRANT MEMBERSHIP IN GROUP 语句
 - 使用, 445
- GRANT 语句
 - DBA 特权, 433
 - RESOURCE 特权, 433
 - WITH GRANT OPTION, 437
 - 创建组, 444
 - 权限, 434
 - 没有口令, 447
 - 组成员资格, 444
 - 表权限, 434
 - 过程, 437
- GrowDB 系统事件
 - 说明, 868
- GrowLog 系统事件
 - 示例, 869
 - 说明, 868
- GrowTemp 系统事件
 - 说明, 868
- GSS-API 库文件

- Kerberos, 107
- Guest 用户
 - 创建, 105
- 概述选项卡
 - SQL Anywhere, 634
- 高度表加密
 - 初始化 [dbinit] 实用程序, 729
- 高度加密
 - ec 服务器选项, 171
 - ek 数据库选项, 238
 - ep 服务器选项, 173
 - AES 算法, 1019
 - DatabaseKey [DBKEY] 连接参数, 259
 - Encryption [ENC] 连接参数, 265
 - Rijndael, 1019
 - Windows Mobile 上的 SQL Anywhere 数据库, 1030
 - 关于, 1031
 - 创建高度加密的数据库, 1020
 - 初始化 [dbinit] 实用程序, 729
 - 卸载 [dbunload] 实用程序, 795
 - 数据库文件, 1019
- 高可用性
 - SQL Anywhere Veritas 聚簇服务器代理, 905
 - 关于, 879
 - 数据库镜像, 880
 - 活动备份, 821
- 高速缓存
 - max_plans_cached 选项, 518
 - 在 Vista 上的 AWE, 39
 - 大小选项, 44
 - 数据库服务器选项, 44
 - 最大大小, 614
 - 服务器名, 129
- 高速缓存大小
 - AWE 的限制, 163
 - 加密数据库问题, 1024
 - 在数据库服务器消息窗口中显示, 165
 - 缺省值, 158
 - 设置, 158
 - 设置最大值, 161
 - 设置最小值, 162
 - 限制, 614
 - 静态, 159
- 高速缓存缓冲区
 - 性能, 158
- 高速缓存预热
 - 用页重装高速缓存, 165
 - 数据库页收集, 160
 - 服务器消息, 166
- 格式
 - 输入文件, 672
- 隔离级别
 - 术语定义, 1100
 - 查询镜像数据库, 898
 - 缺省, 508
 - 设置, 508
- 个人服务器
 - Windows Mobile 上不支持, 336
 - 关于, 36
 - 术语定义, 1100
- 个人服务器示例
 - 运行, 5
- 个人数据库服务器 (见 个人服务器)
- 根登录策略
 - 修改, 417
 - 关于, 416
- 根证书
 - 传送层安全客户端校验, 1040
 - 传送层安全, 1036
- 更改归类
 - 关于, 402
- 更改日志文件设置向导
 - Windows Mobile 上不支持, 339
 - 使用, 16
 - 启动现有数据库的事务日志镜像, 16
- 更新
 - ansi_permissions 选项, 481
 - ansi_update_constraints 选项, 483
 - Interactive SQL 中的值, 654
 - SQL/2003 行为, 483
 - Transact-SQL 权限, 481
 - 检查 SQL Anywhere 更新, 690
- 更新检查程序
 - 关于, 690
- 工具栏
 - Interactive SQL 执行语句按钮, 642
- 工作表
 - 术语定义, 1100
- 功能
 - 生成数据库文档, 635
- 功能键
 - Interactive SQL, 663
- 公共密钥基础结构对象

- 查看, 708
- 公共密钥加密算法
 - 关于, 1031
- 公司名
 - 服务器授权 [dblic] 实用程序, 766
- 公用表表达式
 - max_recursive_iterations 选项, 520
- 公用名
 - certificate_name 协议选项, 288
 - 在 MobiLink 传送层安全校验, 1050
- 共享内存
 - Unix 临时文件配置, 357
 - 保护 Unix 连接安全, 1005
 - 关于, 51
 - 和 CommLinks 连接参数, 254
 - 服务器配置, 221
 - 终端服务, 52
- 估计
 - 恢复时间, 538
 - 行计数, 542
- 故障
 - 恢复, 817
- 故障排除
 - 协议, 143
 - 布线问题, 144
 - 服务器启动, 68
 - 服务器地址, 1065
 - 网络通信, 143
 - 超时, 145
- 故障切换
 - 术语定义, 1100
- 故障转移
 - Veritas 聚簇服务器和 SQL Anywhere, 905
 - 数据库镜像, 880
 - 数据库镜像情形, 902
 - 聚簇, 905
- 固定宽度字符集
 - 关于, 385
- 关闭
 - 指定时间, 214
- 关键连接
 - 术语定义, 1110
- 关键字
 - non_keywords 选项, 523
 - 关闭, 523
- 管理登录策略
 - 关于, 416
 - 只读数据库, 420
- 管理工具
 - Interactive SQL, 636
 - Mac OS X 硬件要求, 621
 - Sybase Central, 620
 - Windows Mobile 上不支持的功能, 336
- 管理连接的用户
 - 关于, 442
- 管理器
 - 关于, 957
- 管理实用程序
 - (参见 数据库管理实用程序)
 - 在 Windows Mobile 上使用, 330
- 管理信息库
 - 关于, 957
- 管理员
 - 监控器用户, 942
- 管理员用户
 - 监控器关于, 942
- 广播转发器实用程序 [dbns11]
 - 使用, 128
 - 语法, 703
- 规范化
 - 术语定义, 1101
- 归类
 - Adaptive Server Enterprise 归类, 406
 - LTM 国际化文件, 1088
 - LTM 字符集问题, 1088
 - LTM 配置文件设置, 1089
 - SQL Anywhere 数据库, 394
 - 关于, 392
 - 创建数据库, 401
 - 初始化期间定制, 729
 - 土耳其语数据库, 411
 - 复制服务器, 1083
 - 多字节, 392
 - 定义, 384
 - 对字符进行排序, 使用, 392
 - 建议在 Unix 平台上使用, 409
 - 建议在 Windows 平台上使用, 408
 - 德语数据库的缺省值, 392
 - 支持的土耳其语归类间的差异, 412
 - 支持的归类列表, 404
 - 更改, 402
 - 替代, 404
 - 术语定义, 1101
 - 确定 CHAR 归类, 399

- 确定 NCHAR 归类, 399
- 确定缺省的, 399
- 缺省, 28
- 选择, 395
- 归类定制
 - dbinit 实用程序, 729
 - Windows Mobile 上的有限支持, 321
 - 关于, 396
 - 瑞典语, 398
 - 选项, 396
- 归类序列
 - 初始化 [dbinit] 实用程序, 729
- 国际语言和字符集
 - 概述, 377
- 国际语言支持
 - 关于, 378
- 过程
 - max_plans_cached 选项, 518
 - SQL Anywhere LTM, 1084
 - 创建权限, 427
 - 在 Interactive SQL 中查找, 648
 - 复制服务器复制, 1085, 1086
 - 安全, 452
 - 权限, 437
 - 生成数据库文档, 635

H

- HasCollationTailoring 属性
 - SQL Anywhere SNMP Extension Agent OID, 984
 - 数据库属性说明, 601
- HasEndianSwapFix 属性
 - SQL Anywhere SNMP Extension Agent OID, 984
 - 数据库属性说明, 601
- HashForcedPartitions 属性
 - 数据库属性说明, 601
 - 连接属性说明, 566
- HashRowsFiltered 属性
 - 数据库属性说明, 601
 - 连接属性说明, 566
- HashRowsPartitioned 属性
 - 数据库属性说明, 601
 - 连接属性说明, 566
- HashWorkTables 属性
 - 数据库属性说明, 601
 - 连接属性说明, 566
- HasNCHARLegacyCollationFix
 - 数据库属性说明, 601

- HasNCHARLegacyCollationFix 属性
 - SQL Anywhere SNMP Extension Agent OID, 984
- HeapsCarver 属性
 - SQL Anywhere SNMP Extension Agent OID, 972
 - 服务器属性说明, 588
 - 连接属性说明, 566
- HeapsLocked 属性
 - SQL Anywhere SNMP Extension Agent OID, 972
 - 服务器属性说明, 588
 - 连接属性说明, 566
- HeapsQuery 属性
 - SQL Anywhere SNMP Extension Agent OID, 972
 - 服务器属性说明, 588
 - 连接属性说明, 566
- HeapsRelocatable 属性
 - SQL Anywhere SNMP Extension Agent OID, 972
 - 服务器属性说明, 588
 - 连接属性说明, 566
- Heimdal Kerberos 客户端
 - Unix 支持, 107
- host 协议选项
 - 使用 IPv6 地址, 135
 - 说明, 292
- HP-UX
 - IPv6 支持, 135
 - SHLIB_PATH 环境变量, 359
- HTML 文件格式
 - Interactive SQL 输出, 680
- HTTP
 - 协议选项, 284
 - 服务器配置, 224
- http_session_timeout 选项
 - SQL Anywhere SNMP Extension Agent OID, 987
 - 说明, 507
 - 连接属性说明, 566
- HttpAddresses 属性
 - SQL Anywhere SNMP Extension Agent OID, 975
 - 服务器属性说明, 588
- HttpNumActiveReq 属性
 - SQL Anywhere SNMP Extension Agent OID, 975
 - 服务器属性说明, 588
- HttpNumConnections 属性
 - SQL Anywhere SNMP Extension Agent OID, 975
 - 服务器属性说明, 588
- HttpNumSessions 属性
 - SQL Anywhere SNMP Extension Agent OID, 975
 - 服务器属性说明, 588

- HttpPorts 属性
 - SQL Anywhere SNMP Extension Agent OID, 975
 - 服务器属性说明, 588
- HTTPS
 - MobiLink 传送层安全, 1051
 - 协议选项, 284
 - 服务器配置, 224
 - 用于 UltraLite 客户端的 MobiLink TLS, 1052
- HttpsAddresses 属性
 - SQL Anywhere SNMP Extension Agent OID, 975
 - 服务器属性说明, 588
- HttpServiceName 属性
 - 连接属性说明, 566
- HttpsNumActiveReq 属性
 - SQL Anywhere SNMP Extension Agent OID, 975
 - 服务器属性说明, 588
- HttpsNumConnections 属性
 - SQL Anywhere SNMP Extension Agent OID, 975
 - 服务器属性说明, 588
- HttpsPorts 属性
 - SQL Anywhere SNMP Extension Agent OID, 975
 - 服务器属性说明, 588
- HTTPS 传送层安全
 - MobiLink , 1051
- 回调函数
 - 数据库服务器, 175
- 函数
 - 使用 SQL Anywhere SNMP Extension Agent 执行, 965
 - 复制服务器复制, 1085
 - 支持 APC 格式, 1085
- 和号
 - 在配置文件中使用时, 695
- 后随字节
 - 关于, 385
 - 连接字符串, 386
- 后台
 - 运行数据库服务器, 57
- 后台启动服务器实用程序 [dbspawn]
 - 语法, 781
 - 退出代码, 781
- 环境变量
 - DYLD_LIBRARY_PATH, 346
 - ERRORLEVEL, 740
 - LD_LIBRARY_PATH, 347
 - LIBPATH, 348
 - ODBC_INI, 350
 - ODBCHOME, 349
 - ODBCINI, 350
 - PATH, 351
 - SACHARSET, 352
 - SADIAGDIR, 353
 - SALANG, 355
 - SALOGDIR, 356
 - SATMP, 357
 - SHLIB_PATH, 359
 - SQLANY11, 360
 - SQLANYSAMP11, 361
 - SQLCONNECT, 362
 - SQLPATH, 363
 - SQLREMOTE, 364
 - SYBASE, 365
 - TEMP, 366
 - TEMPDIR, 366
 - TMP, 366
 - Unix, 344
 - 关于, 344
 - 命令 shell, xv
 - 命令提示符, xv
 - 在 Mac OS X 上设置, 344
 - 在 Unix 上发起, 344
 - 在 Windows Mobile 上设置 TEMP 目录, 374
 - 在 Windows 上设置, 344
 - 设置, 344
 - 连接到数据库实用程序, 123
- 环境变量选项 (*!data* 选项)
 - 缓冲
 - 复制服务器复制命令, 1087
 - 恢复
 - 事务日志, 14, 841
 - 事务日志镜像, 15
 - 介质故障, 833, 839
 - 关于, 817
 - 内部, 852
 - 分布式事务, 213
 - 快速, 821
 - 最小时间, 184
 - 服务器选项, 175
 - 期间的限制, 824
 - 未提交的更改, 834
 - 系统故障, 846
 - 紧急情况, 853
 - 自动, 817
 - 选项, 46

- 针对 Windows Mobile 的策略, 328
- 恢复模式
 - 日志传送管理器 [dbltm] 实用程序, 748
- 恢复日志
 - 关于, 14
- 恢复数据库向导
 - Windows Mobile 上不支持, 339
 - 使用, 835
- 回退日志
 - 关于, 855
 - 术语定义, 1101
- 活动
 - 连接, 212
- 活动备份
 - 与事务日志镜像的区别, 822
 - 关于, 821, 830
 - 备份 [dbbackup] 实用程序, 698
 - 概述, 821
- 活动备份和事务日志镜像之间的区别
 - 关于, 822
- 获取帮助
 - 技术支持, xvi
- 获取验证签名
 - 关于, 70

I

- IANA
 - 替代字符集编码标签, 404
 - 端口号, 303
- IANA 标签
 - 字符集, 408
- iAnywhere.mib 文件
 - 位置, 969
 - 关于, 956, 957
- iAnywhere JDBC 驱动程序
 - 不支持, 336
 - 术语定义, 1101
- iAnywhere Solutions Oracle 驱动程序
 - 创建数据源, 711
- iAnywhere 开发人员社区
 - 新闻组, xvii
- ICU
 - Unicode 归类算法 (UCA), 393
 - Unicode 的国际组件, 381
 - 关于, 381
 - 在 Windows Mobile 上使用, 310
 - 替代字符集编码标签, 404

- 用于字符集转换, 386
- 确定何时需要 ICU, 381
- 语法定制, 393
- ICU 库
 - 创建 Windows Mobile 数据库, 323
 - 卸载 Windows Mobile 上的数据库, 327
 - 避免在 Windows Mobile 上, 324
- identity_password 协议选项
 - dbeng11 -ec, 171
 - dbsrv11 -ec, 171
 - 说明, 294
- IdentitySignature 属性
 - SQL Anywhere SNMP Extension Agent OID, 984
 - 数据库属性说明, 601
- identity 协议选项
 - 说明, 294
- IdleCheck 属性
 - SQL Anywhere SNMP Extension Agent OID, 980
 - 数据库属性说明, 601
- IdleChkpt 属性
 - SQL Anywhere SNMP Extension Agent OID, 980
 - 数据库属性说明, 601
- IdleChkTime 属性
 - SQL Anywhere SNMP Extension Agent OID, 980
 - 数据库属性说明, 601
- IdleTimeout 属性
 - SQL Anywhere SNMP Extension Agent OID, 975
 - 服务器属性说明, 588
 - 连接属性说明, 566
- IdleTime 事件
 - 轮询, 873
- IdleWrite 属性
 - SQL Anywhere SNMP Extension Agent OID, 980
 - 数据库属性说明, 601
- Idle 连接参数
 - 说明, 268
- IndAdd 属性
 - SQL Anywhere SNMP Extension Agent OID, 980
 - 数据库属性说明, 601
 - 连接属性说明, 566
- IndLookup 属性
 - SQL Anywhere SNMP Extension Agent OID, 980
 - 数据库属性说明, 601
 - 连接属性说明, 566
- InfoMaker
 - 术语定义, 1101
- InitString 连接参数

-
- ODBC 连接参数说明, 714
 - INI 文件
 - 关于, 373
 - 用 dbfhide 实现简单加密, 724
 - input_format 选项
 - Interactive SQL 设置, 667
 - 说明, 672
 - INPUT 语句
 - 在 Interactive SQL 中插入新行, 656
 - INSERT 权限
 - 关于, 428
 - 授予, 434
 - INSERT 语句
 - LTM 支持的操作, 1084
 - 在 Interactive SQL 中生成, 649
 - 字符串的截断, 549
 - install-dir
 - 文档用法, xiv
 - INSTALL JAVA 语句
 - Windows Mobile 上不支持, 337
 - installulnet.exe
 - 在 Vista 上可能需要权限提升, 38
 - integrated_server_name 选项
 - SQL Anywhere SNMP Extension Agent OID, 987
 - 指定集成登录的域控制器服务器, 102
 - 说明, 507
 - 连接属性说明, 566
 - Integrated 连接参数
 - 说明, 269
 - Interactive SQL
 - (参见 dbisql 实用程序)
 - (参见 Interactive SQL 实用程序 [dbisql])
 - .sql 文件的缺省编辑器, 643
 - 重新调用命令, 645
 - dbisql 语法, 740
 - Interactive SQL 选项按字母顺序排序的列表, 667
 - Mac OS X 硬件要求, 636
 - SQL 语句, 662
 - SQL 语句按字母顺序排序的列表, 662
 - 中断命令, 647
 - 为读取和写入文件指定代码页, 671
 - 停止命令, 647
 - 关于, 636
 - 创建 Windows Mobile 数据库, 325
 - 删除行, 656
 - 功能键, 663
 - 单步执行 SQL 语句, 641
 - 取消命令, 647
 - 只执行 [SQL 语句] 窗格中选定的文本, 643
 - 启动, 637
 - 命令历史记录窗口, 645
 - 命令行, 741
 - 图形式计划, 652
 - 复制行, 656
 - 实用程序, 740
 - 对结果集排序, 657
 - 打印, 653
 - 打开多个窗口, 657
 - 打开源代码控制项目, 660
 - 执行命令, 641
 - 执行命令文件, 643
 - 执行多条语句, 643
 - 报告的错误, 647
 - 插入行, 655
 - 收藏夹, 644
 - 文本完成, 683
 - 显示数据, 641
 - 显示查询编辑器, 663
 - 更新计算列, 655
 - 术语定义, 1102
 - 查找表、列和过程, 648
 - 查看数据时的意外符号, 387
 - 源代码控制集成, 658
 - 禁用表编辑, 655
 - 签入文件, 662
 - 签出文件, 661
 - 管理 Windows Mobile 上的数据库, 333
 - 编辑表值, 654
 - 行号, 639
 - 记录命令, 647
 - 设置选项, 667
 - 语法, 740
 - 软件更新, 690
 - 连接到数据库, 637
 - 选项, 640, 667
 - 通过经过验证的应用程序使用, 70
 - 配置快速启动程序, 686
 - 配置源代码控制, 658
 - 键盘快捷方式, 663
 - Interactive SQL 实用程序 [dbisql]
 - (参见 dbisql 实用程序)
 - (参见 Interactive SQL)
 - 支持的平台, 742
 - 语法, 740
-

- 退出代码, 742
- Interactive SQL 选项
 - auto_commit, 668
 - auto_refetch, 669
 - bell, 669
 - command_delimiter, 670
 - commit_on_exit, 671
 - default_isql_encoding, 671
 - echo, 672
 - input_format, 672
 - Interactive SQL 选项按字母顺序排序的列表, 667
 - isql_allow_read_client_file, 673
 - isql_command_timing, 674
 - isql_escape_character, 675
 - isql_field_separator, 676
 - isql_maximum_displayed_rows, 677
 - isql_print_result_set, 677
 - isql_quote, 678
 - isql_show_multiple_result_sets, 678
 - nulls, 679
 - on_error, 679
 - output_format, 680
 - output_length, 681
 - output_nulls, 681
 - sql_allow_write_client_file, 674
 - truncation_length, 682
 - 分类, 466
 - 初始设置, 465
 - 设置, 667
- Internet SCSI
 - 存储数据库文件, 14
- INT 连接参数
 - 说明, 269
- IN 关键字
 - CREATE TABLE 语句, 23
- IOParallelism 属性
 - SQL Anywhere SNMP Extension Agent OID, 984
 - 数据库属性说明, 601
- IOToRecover 属性
 - SQL Anywhere SNMP Extension Agent OID, 980
 - 数据库属性说明, 601
- IPv4
 - 关于, 135
 - 排除连接故障, 143
- IPv6
 - 与 BCAST 协议选项一起使用, 135
 - 与 Broadcast 协议选项一起使用, 135
 - 与 host 协议选项一起使用, 135
 - 与 IP 协议选项一起使用, 135
 - 与 ME 协议选项一起使用, 135
 - 与 MyIP 协议选项一起使用, 135
 - 关于, 135
 - 排除连接故障, 144
 - 接口名称, 135
 - 接口标识符, 135
 - 支持的平台, 135
- IP 地址
 - ping, 143
 - 确定 Windows Mobile 设备, 318
 - 配置 Open Server, 1064
- IP 协议选项
 - 使用 IPv6 地址, 135
 - 说明, 292
- IQStore 属性
 - SQL Anywhere SNMP Extension Agent OID, 984
- iSCSI
 - 存储数据库文件, 14
- IsDebugger 属性
 - 连接属性说明, 566
- IsEccAvailable 属性
 - SQL Anywhere SNMP Extension Agent OID, 975
 - 服务器属性说明, 588
- IsFipsAvailable 属性
 - SQL Anywhere SNMP Extension Agent OID, 975
 - 服务器属性说明, 588
- IsIQ 属性
 - SQL Anywhere SNMP Extension Agent OID, 975
- IsNetworkServer 属性
 - SQL Anywhere SNMP Extension Agent OID, 975
 - 服务器属性说明, 588
- isolation_level 选项
 - Open Client, 1067
 - SQL Anywhere SNMP Extension Agent OID, 987
 - Transact-SQL 兼容性, 473
 - 说明, 508
 - 连接属性说明, 566
- IsolationLevel 连接参数
 - ODBC 连接参数说明, 714
- isql_allow_read_client_file 选项
 - Interactive SQL 设置, 667
 - 说明, 673
- isql_allow_write_client_file 选项
 - Interactive SQL 设置, 667
 - 说明, 674

- isql_command_timing 选项
 - Interactive SQL 设置, 667
 - 说明, 674
- isql_escape_character 选项
 - Interactive SQL 设置, 667
 - 复制数据, 656
 - 说明, 675
- isql_field_separator 选项
 - Interactive SQL 设置, 667
 - 复制数据, 656
 - 说明, 676
- isql_maximum_displayed_rows 选项
 - Interactive SQL 设置, 667
 - 说明, 677
- isql_print_result_set 选项
 - Interactive SQL 设置, 667
 - 说明, 677
- isql_quote 选项
 - Interactive SQL 设置, 667
 - 复制数据, 656
 - 说明, 678
- isql_show_multiple_result_sets 选项
 - Interactive SQL 设置, 667
 - 说明, 678
- IsRsaAvailable 属性
 - SQL Anywhere SNMP Extension Agent OID, 975
 - 服务器属性说明, 588
- IsRuntimeServer 属性
 - SQL Anywhere SNMP Extension Agent OID, 975
 - 服务器属性说明, 588
- IsService 属性
 - SQL Anywhere SNMP Extension Agent OID, 975
 - 服务器属性说明, 588
- J**
- JAR 文件
 - 术语定义, 1102
- Java
 - cp 服务器选项, 164
 - java_location 选项, 509
 - java_main_userid, 510
 - java_vm_options, 510
 - 替代字符集编码标签, 404
 - 连接参数, 81
- java_location 选项
 - SQL Anywhere SNMP Extension Agent OID, 987
 - 说明, 509
 - 连接属性说明, 566
- java_main_userid 选项
 - SQL Anywhere SNMP Extension Agent OID, 987
 - 说明, 510
 - 连接属性说明, 566
- java_vm_options 选项
 - SQL Anywhere SNMP Extension Agent OID, 987
 - 说明, 510
 - 连接属性说明, 566
- JavaVM 属性
 - SQL Anywhere SNMP Extension Agent OID, 984
 - 数据库属性说明, 601
- Java 类
 - 术语定义, 1102
- java 目录
 - 关于, 368
- jConnect
 - Kerberos 验证, 110
 - TDS, 1058
 - Windows Mobile, 322
 - Windows Mobile 上的受限制功能, 336
 - 初始化 [dbinit] 实用程序, 729
 - 升级 [dbupgrad] 实用程序, 808
 - 术语定义, 1102
- jConnect 元数据支持
 - Windows Mobile, 322
- JDBC
 - ASE 兼容性选项, 473
 - 术语定义, 1102
- 校验
 - Sybase Central 中的数据库, 860
 - Sybase Central 中的表, 860
 - 事务日志, 851
 - 备份, 832, 858
 - 执行权限, 427
 - 数据库, 810, 832, 858
 - 术语定义, 1104
- 校验表
 - 从 Sybase Central, 860
- 校验服务器
 - SQL Anywhere 传送层安全, 1043
- 校验和
 - 校验 [dbvalid] 实用程序, 810
 - 为 Windows Mobile 数据库启用, 322
 - 关于, 859
 - 初始化 [dbinit] 实用程序, 729
 - 术语定义, 1104

- 自动为 Windows Mobile 数据库启用, 859
- 校验校验和
 - 关于, 859
- 校验 MobiLink 服务器
 - MobiLink 传送层安全, 1049
- 校验事务日志
 - 关于, 851
- 校验实用程序 [dbvalid]
 - 退出代码, 812
- 校验数据库
 - 从 Sybase Central, 860
 - 提高性能, 862
- 校验数据库向导
 - 使用, 860
- 校证书字字段
 - MobiLink 传送层安全, 1049
 - SQL Anywhere 传送层安全, 1044
- 基表
 - 术语定义, 1102
- 基于 SQL 的同步
 - 术语定义, 1103
- 基于会话的同步
 - 术语定义, 1102
- 基于脚本的上载
 - 术语定义, 1103
- 基于文件的下载
 - 术语定义, 1103
- 集成登录
 - integrated_server_name 选项, 507
 - login_mode 选项, 512
 - Windows 用户组, 102
 - 使用, 99, 102
 - 关于, 99
 - 创建, 100
 - 启用, 99
 - 安全功能, 114, 1006
 - 安全问题, 105
 - 撤销权限, 101
 - 操作系统, 99
 - 术语定义, 1103
 - 禁止连接, 103
 - 缺省用户, 105
 - 网络方面, 104
- 技术支持
 - 为技术支持加密数据库, 1022
 - 新闻组, xvii
 - 解密数据库为, 1023
- 计划
 - max_plans_cached 选项, 518
 - 备份, 844
 - 备份和恢复, 843
 - 捕获最近, 230
 - 控制优化程序的使用, 529
- 计划查看器
 - 关于, 651
 - 浏览, 651
- 计算列
 - 卸载数据库时重新计算, 795
 - 在 Interactive SQL 中重新计算, 655
 - 卸载数据库, 805
 - 在 Interactive SQL 中更新, 655
 - 在 Interactive SQL 中添加到新行, 655
- 记录
 - HTTP 客户端信息, 229
 - Interactive SQL 中的命令, 647
 - 事务日志, 14
 - 数据库服务器操作, 40
 - 数据库服务器消息, 196
- 记录 SQL 语句
 - 关于, 42
- 记录数据库
 - 关于, 635
- 继承
 - 权限, 422
 - 特权, 422
 - 登录策略选项, 416
- 加密
 - (参见 加密)
 - ec 服务器选项, 171
 - ek 服务器选项, 238
 - ep 服务器选项, 173
 - es 服务器选项, 174
 - AES 算法, 1019
 - certificate_company 协议选项, 287
 - certificate_name 协议选项, 288
 - certificate_unit 协议选项, 289
 - Encryption [ENC] 连接参数, 265
 - FIPS, 1033
 - INI 文件, 724
 - MobiLink, 1047
 - TDS 协议选项, 305
 - trusted_certificates 协议选项, 307
 - Windows Mobile, 322

Windows Mobile 上的 SQL Anywhere 数据库, 1029

Windows Mobile 上的客户端/服务器通信, 1030

使用 dbinit 创建数据库, 729

关于, 1019

列, 1024

创建后的表, 1027

创建时的表, 1027

加密属性, 601

加密数据库的性能, 1024

口令, 264, 1006

启用, 1027

数据库, 1020

数据库文件, 1019

数据库文件, 创建后, 1021

文件隐藏 [dbfhide] 实用程序, 724

比较 CREATE ENCRYPTED FILE 和 CREATE ENCRYPTED DATABASE, 1022

端对端, 1047

简单, 1019

表, 1026, 1027

解密数据库, 1022

通信, 1031

针对技术支持的数据库, 1022

高度, 171, 173, 174, 238, 259, 265, 1019

加密 MobiLink 客户端/服务器通信

关于, 1047

加密密钥

DBKEY 连接参数, 259

事务日志 [dblog] 实用程序, 792

使用 CREATE ENCRYPTED DATABASE 语句更改, 1023

保护, 1023

初始化 [dbinit] 实用程序, 729

卸载 [dbunload] 实用程序, 795

日志传送管理器 [dbltm] 实用程序, 748

消除 [dberase] 实用程序, 722

翻译日志文件 [dbtran] 实用程序, 753

选择, 1023

加密算法

AES, 1019

Rijndael, 1019

公共密钥, 1031

监控

已登录用户, 780

性能, 使用直方图, 726

数据库, 913

数据库选项设置, 465

监控器, 913

监控器

(参见 MobiLink 监控器)

(参见 性能监控器)

(参见 监控器度量)

[度量] 选项卡, 929

使用封锁停止监控, 939

使用户与资源相关联, 943

修复资源, 940

停止监控器, 924

停止监控资源, 938

关于, 913

创建用户, 942

删除安装的对象, 950

删除度量, 933

删除用户, 944

删除警告, 947

删除资源, 940

启用电子邮件通知, 948

在单独的计算机上启动, 923

在单独的计算机上安装, 951

在生产环境中运行, 914

安全, 944

安装的对象, 950

封锁, 939

度量, 928

开始监控资源, 935

快速入门, 917

手工停止监控, 939

收集间隔, 936

教程, 918

断开连接, 926

本地启动, 923

添加资源, 935

状态, 928

用户类型, 942

电子邮件通知, 947

疑难解答, 952

监控资源, 935

管理员用户, 942

网络配置, 915

要求, 914

要求登录, 944

解析警告, 946

警告, 946

警告错误报告, 949

- 资源, 927, 935
- 连接, 925
- 退出, 924
- 选项卡, 929
- 错误报告, 949
- 限制, 914
- 监控器度量
 - [CPU] 选项卡, 931
 - [HTTP] 选项卡, 932
 - [度量] 选项卡, 929
 - [磁盘] 选项卡, 932
 - [警告] 选项卡, 930
 - 未调度的请求选项卡, 931
 - 内存选项卡, 931
 - 删除, 933
 - 失败的连接选项卡, 933
 - 当在指定时间内发生相同情况时取消发出警告, 937
 - 指定要收集的度量, 936
 - 收集间隔, 936
 - 服务器选项卡, 930
 - 查询选项卡, 933
 - 警告, 937
 - 连接选项卡, 932
 - 镜像选项卡, 933
- 监视数据库健康和统计信息
 - 关于, 634
- 监视项目列表
 - 数据库选项, 465
- 监听器
 - 术语定义, 1103
- 兼容性
 - ANSI, 472
 - SQL, 472
 - Transact-SQL, 472
- 兼容性选项
 - allow_nulls_by_default, 477
 - ansi_blanks, 480
 - ansi_close_cursors_on_rollback, 481
 - ansi_permissions, 481
 - ansi_substring, 482
 - ansi_update_constraints, 483
 - ansinull, 484
 - ASE 兼容性选项, 473
 - chained, 488
 - close_on_endtrans, 489
 - continue_after_raiseerror, 492
 - conversion_error, 493
 - escape_character, 502
 - fire_triggers, 504
 - isolation_level, 508
 - non_keywords, 523
 - on_tsq_error, 528
 - quoted_identifier, 537
 - sql_flagger_error_level, 545
 - sql_flagger_warning_level, 546
 - string_rtruncation, 549
 - time_format, 553
 - timestamp_format, 554
 - Transact-SQL 兼容性选项, 473
 - tsql_outer_joins, 557
 - tsql_variables, 557
 - 分类, 466
 - 初始设置, 465
 - 按字母顺序排序的数据库兼容性选项列表, 472
- 检查点
 - checkpoint_time 选项, 488
 - 之后删除事务日志, 194
 - 关于, 19
 - 备份, 852
 - 备份期间不允许, 824
 - 术语定义, 1103
 - 紧急情况, 853
 - 间隔, 178
- 检查点日志
 - 关于, 17
- 检查更新
 - 关于, 690
- 检查软件更新
 - 关于, 690
- 检索
 - Interactive SQL 中的命令, 645
- 简单加密
 - Windows Mobile 上的 SQL Anywhere 数据库, 1029
 - 关于, 1019
- 简单网络管理协议 (SNMP)
- 键盘快捷方式
 - Interactive SQL, 663
 - Sybase Central, 625
 - 代码编辑器, 626
 - 文本完成, 684
- 键盘映射
 - 关于, 384

键入完成 (见 文本完成)

健康和统计信息
查看, 634
监控器, 913

将客户端设置为信任证书
传送层安全, 1040

脚本
术语定义, 1103

脚本版本
术语定义, 1104

脚本执行 [dbrunsql] 实用程序
语法, 761

角色
数据库镜像, 880
术语定义, 1104

角色名
术语定义, 1104

角色切换
数据库镜像, 880

教程
从 Sybase Central 运行 Windows Mobile 数据库, 330
使用 Interactive SQL 管理 Windows Mobile 数据库, 333
在 Windows Mobile 上运行数据库服务器, 330
复制服务器, 1073
多个数据库的数据库镜像, 891
数据库镜像, 887
监控器, 918
连接到示例数据库, 3

接口
数据库服务器, 5

接口标识符
IPv6 地址, 135
Linux 上必需, 135

接口库
查找, 124
连接, 79

接口名称
IPv6 地址, 135

接口文件
Open Server, 1074
日志传送管理器 [dbltn] 实用程序, 748
配置, 1062

截断
字符串, 549

结果
查看数据时的意外符号, 387

结果集
删除行, 656
在 Interactive SQL 中禁用编辑表值, 655
在 Interactive SQL 中编辑表值, 654
复制行, 656
打印, 653
排序, 657
插入行, 655

解密
数据库, 1022
表, 1026
针对技术支持的数据库, 1023

介质故障
事务日志, 841
保护, 846
关于, 846
恢复, 833, 839

仅检查点模式
数据库服务器, 188

进程并发水平
提高, 50
数据库服务器, 50
选择, 51
降低, 51

禁用
在 Interactive SQL 中进行表编辑, 655

禁用表编辑
关于, 654

禁用数据库功能
-sf 服务器选项, 207
使用安全功能密钥, 544
关于, 1010
指定安全功能密钥, 210

经过验证的连接
connection_authentication 选项, 491

经过验证的数据库
database_authentication 选项, 495
升级, 74

经过验证的应用程序
connection_authentication 选项, 491
database_authentication 选项, 495
关于, 70
开发, 70
数据库验证, 71
编程接口示例, 72
配置, 71

- 验证签名, 70
 - 经过准备的语句数
 - max_statement_count 选项, 520
 - 服务器支持的最大值, 614
 - 井号
 - 在配置文件中使用, 695
 - 警告
 - 监控器, 946
 - 监控器取消, 949
 - 监控器电子邮件通知, 947
 - 静默
 - 数据库服务器, 46
 - 镜像
 - (参见 数据库镜像)
 - sm 选项, 241
 - sn 选项, 242
 - xa 选项, 222
 - xf 选项, 223
 - xp 选项, 244
 - SQL Anywhere SNMP Extension Agent 陷阱, 966
 - 事务日志, 14, 15, 16
 - 从主服务器故障恢复, 900
 - 仲裁服务器角色, 883
 - 优点, 882
 - 停止数据库服务器, 900
 - 关于, 880
 - 创建具有事务日志镜像的数据库, 22
 - 同步模式, 883, 884
 - 同步状态, 885
 - 备份, 902
 - 客户端连接, 896
 - 异步模式, 884
 - 异步满页模式, 884
 - 性能, 902
 - 情形, 902
 - 教程, 887, 891
 - 状态信息文件, 885
 - 确定主服务器, 897
 - 系统事件, 901
 - 设置 synchronize_mirror_on_commit 选项, 551
 - 配置, 896
 - 限制, 882
 - 镜像服务器
 - 停止, 900
 - 只读数据库访问, 898
 - 数据库镜像概述, 880
 - 镜像日志
 - 术语定义, 1104
 - 镜像事务日志 (见 事务日志镜像)
 - 镜像数据库
 - 只读访问, 898
 - 查询, 898
 - 镜像系统
 - 关于, 880
 - 局部临时表
 - 术语定义, 1104
 - 聚簇
 - 关于, 905
 - 聚簇散列分组依据
 - optimization_workload 选项, 530
- ## K
- KeepaliveTimeout 协议选项
 - 说明, 295
 - Kerberos
 - kl 选项, 190
 - kr 选项, 191
 - krb 选项, 192
 - CSFC5KTNAME 环境变量, 107
 - GSS-API 库文件, 107
 - jConnect 连接, 110
 - Kerberos 主体, 108
 - Kerberos 连接参数 [KRB], 270
 - keytab 文件, 107
 - KRB5_KTNAME 环境变量, 107
 - login_mode 选项, 512
 - Open Client 连接, 110
 - Windows Mobile 上不支持, 336
 - 临时选项, 115
 - 关于, 107
 - 在 Windows 上使用 SSPI, 112
 - 安全问题, 105
 - 客户端, 107
 - 密钥分发中心, 108
 - 授予权限, 111
 - 撤销权限, 111
 - 票据授予票据, 110
 - 连接疑难解答, 113
 - Kerberos 登录
 - (参见 Kerberos)
 - Kerberos 连接参数
 - 说明, 270
 - Kerberos 密钥分发中心
 - 关于, 108

Kerberos 主体
 关于, 108

KeysInSQLStatistics 连接参数
 ODBC 连接参数说明, 714

keytab 文件
 缺省位置, 107

KRB5_KTNAME 环境变量
 Kerberos, 107

KRB 连接参数
 说明, 270

KTO 协议选项
 说明, 295

开发人员社区
 新闻组, xvii

开发应用程序
 关于, 70

开关
 dbisqlc 实用程序, 720
 Linux 服务 [dbsvc] 实用程序, 769
 MobiLink 证书创建 [createcert], 705
 MobiLink 证书查看器 [viewcert] 实用程序, 708
 SQL Anywhere 控制台 [dbconsole] 实用程序, 780
 SQL Anywhere 脚本执行 [dbrunsql] 实用程序, 761
 viewcert 实用程序, 708
 Windows 服务 [dbsvc] 实用程序, 773
 停止服务器 [dbstop] 实用程序, 783
 后台启动服务器 [dbspawn] 实用程序, 781
 广播转发器 [dbns11] 实用程序, 703
 支持 [dbsupport] 语法, 785
 日志传送管理器实用程序 [dbltn] 实用程序, 748
 服务器授权 [dblic] 实用程序, 766
 服务器枚举 [dblocate] 实用程序, 763
 语言选择 [dblang] 实用程序, 745

可变宽度字符集
 关于, 385

可用性
 数据库服务器, 57
 高, 821

可执行文件
 位置, 368
 针对 Vista 已签名, 39

可执行文件名
 服务器授权 [dblic] 实用程序, 766

客户端
 DatabaseKey [DBKEY] 连接参数, 259
 Encryption [ENC] 连接参数, 265
 Kerberos, 107
 启动支持传送层安全的 SQL Anywhere, 1043
 备份, 830
 备份快速入门, 818
 标识, 249
 连接到镜像数据库, 896
 配置为信任证书, 1040

客户端/服务器
 SQL 语句, 387
 字符集转换, 387
 术语定义, 1104

客户端/服务器通信
 语言问题, 384

客户端安全 trusted_certificates 协议选项
 MobiLink 传送层安全, 1050

客户端文件
 allow_read_client_file 选项, 478
 allow_write_client_file 选项, 479
 isql_allow_read_client_file 选项 [Interactive SQL], 673
 isql_allow_write_client_file 选项 [Interactive SQL], 674
 read_client_file 受保护的功能, 207
 READCLIENTFILE 特权, 426
 write_client_file 受保护的功能, 207
 WRITECLIENTFILE 特权, 427

客户端消息存储库
 术语定义, 1104

客户端消息存储库 ID
 术语定义, 1105

客户端语句高速缓存
 关于, 516

空格
 ANSI 行为, 480
 连接字符串, 80

空格填充
 关于, 729
 初始化 [dbinit] 实用程序, 729

空闲服务器
 事件示例, 870

空值
 为导出定义, 681

控制台 (见 SQL Anywhere 控制台实用程序) (见 SQL Anywhere 监控器控制台) (见 数据库服务器消息窗口)
 控制台实用程序 [dbconsole]

- 使用, 687
- 语法, 780
- 配置, 688
- 控制线程行为
 - 关于, 48
- 控制用户可以执行的任务
 - 关于, 1009
- 口令
 - Interactive SQL 命令历史记录, 645
 - LTM 配置文件, 750
 - NEWPWD 连接参数, 274
 - post_login_procedure 选项, 532
 - PWD 连接参数, 276
 - SQL Remote 保存, 543
 - verify_password_function 选项, 560
 - 为实用程序数据库设置, 211
 - 关于, 431
 - 到期, 274
 - 加密, 264
 - 安全功能, 1006
 - 安全提示, 1005
 - 安装的对象, 950
 - 实用程序数据库, 29
 - 更改, 432
 - 最大长度, 614
 - 最小长度, 522
 - 登录策略, 416
 - 监控器用户, 942
 - 缺省, 425
 - 长度, 1005
 - 验证, 1007
- 口令参数
 - LDAP, 139
- 口令到期
 - NewPassword 连接参数, 274
- 口令规则
 - login_procedure, 513
 - verify_password_function 选项, 560
- 口令验证
 - 关于, 1007
- 库
 - DYLD_LIBRARY_PATH 环境变量 [Mac OS X], 346
 - Kerberos GSS-API 库文件, 107
 - Kerberos 验证, 107
 - LD_LIBRARY_PATH 环境变量 [Linux 和 Solaris], 347
 - LIBPATH 环境变量 [AIX], 348
 - SHLIB_PATH 环境变量 [HP-UX], 359
 - Windows Mobile 上的 ICU 需要, 310
 - 为 dbping 实用程序装载, 757
 - 查找接口库, 124
- 快捷方式
 - Interactive SQL, 663
 - Sybase Central, 625
- 快速启动程序
 - 关于, 686
 - 更改语言设置, 746
- 快速入门
 - 传送层安全, 1035
 - 备份, 818
- 快照
 - 时间点, 829
- 快照隔离
 - isolation_level 数据库选项, 508
 - updatable_statement_isolation 选项, 557
 - 术语定义, 1105
- 框架类型
 - 关于, 144
- 扩展字符
 - 关于, 385
- L**
- LANalyzer
 - 网络通信故障排除, 144
- Language 连接参数
 - 说明, 271
- Language 属性
 - SQL Anywhere SNMP Extension Agent OID, 975, 984
 - 数据库属性说明, 601
 - 服务器属性说明, 588
 - 连接属性说明, 566
- LANG 连接参数
 - 说明, 271
- LastConnectionProperty 属性
 - SQL Anywhere SNMP Extension Agent OID, 975
 - 服务器属性说明, 588
- LastDatabaseProperty 属性
 - SQL Anywhere SNMP Extension Agent OID, 975
 - 服务器属性说明, 588
- LastIdle 属性
 - 连接属性说明, 566
- LastOption 属性

-
- SQL Anywhere SNMP Extension Agent OID, 975
 - 服务器属性说明, 588
 - LastPlanText 属性
 - 连接属性说明, 566
 - LastReqTime 属性
 - 连接属性说明, 566
 - LastServerProperty 属性
 - SQL Anywhere SNMP Extension Agent OID, 975
 - 服务器属性说明, 588
 - LastStatement 属性
 - 连接属性说明, 566
 - LazyAutocommit 连接参数
 - ODBC 连接参数说明, 714
 - LazyClose 连接参数
 - 说明, 272
 - LCLOSE 连接参数
 - 说明, 272
 - LD_LIBRARY_PATH 环境变量
 - 说明, 347
 - LDAP 协议选项
 - 说明, 295
 - LDAP 验证
 - AIX, 138
 - LDAP 协议选项, 295
 - Windows Mobile 上不支持, 336
 - 服务器枚举 [dblocate] 实用程序, 765
 - 连接与, 138
 - LegalCopyright 属性
 - SQL Anywhere SNMP Extension Agent OID, 975
 - 服务器属性说明, 588
 - LegalTrademarks 属性
 - SQL Anywhere SNMP Extension Agent OID, 975
 - 服务器属性说明, 588
 - LF 协议选项
 - 说明, 297
 - libctl.cfg 文件
 - DSEdit, 1063
 - LIBPATH 环境变量
 - 说明, 348
 - LicenseCount 属性
 - SQL Anywhere SNMP Extension Agent OID, 975
 - 服务器属性说明, 588
 - LicensedCompany 属性
 - SQL Anywhere SNMP Extension Agent OID, 975
 - 服务器属性说明, 588
 - LicensedUser 属性
 - SQL Anywhere SNMP Extension Agent OID, 975
 - 服务器属性说明, 588
 - LicenseType 属性
 - SQL Anywhere SNMP Extension Agent OID, 975
 - 服务器属性说明, 588
 - LINKS 连接参数
 - 圆括号, 386
 - 说明, 254
 - 选项, 51
 - Linux
 - IPv6 地址所需的接口标识符, 135
 - IPv6 支持, 135
 - LD_LIBRARY_PATH 环境变量, 347
 - SELinux 策略, 1004
 - 使用 [服务器启动选项] 窗口, 219
 - 启动 dbconsole, 687
 - 启动 Interactive SQL, 637
 - 在 shell 模式下查看服务器消息, 215, 217
 - 查看数据库服务器消息窗口, 219
 - 禁用异步 I/O, 215
 - 线程行为, 48
 - Linux 服务
 - 数据库服务器, 57
 - LivenessTimeout 连接参数
 - 说明, 272
 - LivenessTimeout 属性
 - SQL Anywhere SNMP Extension Agent OID, 975
 - 服务器属性说明, 588
 - 连接属性说明, 566
 - LOAD TABLE 语句
 - 安全, 1018
 - 限制用于数据库镜像, 882
 - localhost 计算机名
 - 配置 Open Server, 1064
 - LocalOnly 协议选项
 - 说明, 296
 - LocalSystem 帐户
 - 关于, 59
 - 选项, 63
 - LOCAL 协议选项
 - 说明, 296
 - Location 注册表条目
 - Windows 上的文件搜索, 370
 - lock_rejected_rows 选项
 - SQL Anywhere SNMP Extension Agent OID, 987
 - 连接属性说明, 566
 - LockCount 属性
 - SQL Anywhere SNMP Extension Agent OID, 980

- 数据库属性说明, 601
- 连接属性说明, 566
- LockedCursorPages 属性
 - SQL Anywhere SNMP Extension Agent OID, 972
 - 服务器属性说明, 588
- LockedHeapPages 属性
 - SQL Anywhere SNMP Extension Agent OID, 972
 - 服务器属性说明, 588
- LockIndexID 属性
 - 连接属性说明, 566
- LockName 属性
 - 连接属性说明, 566
- LockRowID 属性
 - 连接属性说明, 566
- LockTableOID 属性
 - 连接属性说明, 566
- LockTablePages 属性
 - SQL Anywhere SNMP Extension Agent OID, 980
 - 数据库属性说明, 601
- log_deadlocks 选项
 - SQL Anywhere SNMP Extension Agent OID, 987
 - 说明, 511
 - 连接属性说明, 566
- LogDiskSpace 系统事件
 - 说明, 868
- LogFileFragments 属性
 - SQL Anywhere SNMP Extension Agent OID, 984
 - 数据库属性说明, 601
- Logfile 连接参数
 - 说明, 273
- LogFile 协议选项
 - 说明, 297
- LogFormat 协议选项
 - 说明, 297
- LogFreeCommit 属性
 - SQL Anywhere SNMP Extension Agent OID, 980
 - 数据库属性说明, 601
 - 连接属性说明, 566
- login_mode 选项
 - SQL Anywhere SNMP Extension Agent OID, 987
 - 说明, 512
 - 连接属性说明, 566
 - 集成登录, 99
- login_procedure 选项
 - SQL Anywhere SNMP Extension Agent OID, 987
 - 使用 RAISERROR 禁止连接, 513
 - 实现口令到期功能, 1007
 - 说明, 513
 - 连接属性说明, 566
- LoginTime 属性
 - 连接属性说明, 566
- LogMaxSize 协议选项
 - 说明, 298
- LogMirrorName 属性
 - SQL Anywhere SNMP Extension Agent OID, 984
 - 数据库属性说明, 601
- LogName 属性
 - SQL Anywhere SNMP Extension Agent OID, 984
 - 数据库属性说明, 601
- LogOptions 协议选项
 - 说明, 299
- LogWrite 属性
 - SQL Anywhere SNMP Extension Agent OID, 980
 - 数据库属性说明, 601
 - 连接属性说明, 566
- LOG 连接参数
 - 说明, 273
- LOG 协议选项
 - 说明, 297
- LOPT 协议选项
 - 说明, 299
- LSIZE 协议选项
 - 说明, 298
- LTM
 - (参见 LTM 实用程序)
 - Open Client/Open Server 字符集, 1088
 - Open Client/Open Server 归类, 1088
 - 事务日志管理, 1090
 - 事务日志选项, 794
 - 启动, 1079
 - 字符集, 1088
 - 字符集配置, 1089
 - 归类, 1088, 1089
 - 接口文件, 748
 - 支持的操作, 1084
 - 术语定义, 1106
 - 配置, 1074, 1086
 - 配置文件, 749, 1079
- LTM_admin_pw 参数
 - LTM 配置文件, 750
 - 启动 LTM, 1079
- LTM_admin_user 参数
 - LTM 配置文件, 750
 - 启动 LTM, 1079

LTM_charset 参数
 LTM 配置文件, 750, 1089
 启动 LTM, 1079

LTM_language 参数
 LTM 配置文件, 1089

LTM_sortorder 参数
 LTM 配置文件, 1089

LTMGeneration 属性
 SQL Anywhere SNMP Extension Agent OID, 984
 数据库属性说明, 601

LTMTrunc 属性
 SQL Anywhere SNMP Extension Agent OID, 984
 数据库属性说明, 601

LTM 配置文件
 关于, 1086
 创建, 1087
 字符集, 1089
 格式, 1087

LTM 实用程序
 (参见 LTM)
 标识符, 1083
 组件, 1071
 语法, 748

LTO 连接参数
 说明, 272

离开
 组, 445

联邦信息处理标准
 关于, 1033

联合
 查询编辑器不支持, 651

联机备份
 关于, 817, 819

联机手册
 PDF, xii

连接
 ADO, 97
 BroadcastListener [BLISTENER] 协议选项, 286
 ClientPort [CPORT] 协议选项, 290
 dblocate 和 LDAP, 765
 dblocate 疑难解答, 763
 dbping 疑难解答, 757
 dedicated_task 选项, 499
 HOST [IP] 协议选项, 292
 Interactive SQL, 130
 LDAP 协议选项, 295
 OLE DB, 97
 RAS, 137
 ServerPort [PORT] 协议选项, 303
 Sybase Central 连接配置文件, 89
 Windows Mobile, 318
 Windows Mobile 和 ODBC 数据源, 94
 Windows Mobile 数据库和桌面应用程序, 85
 为主站点创建, 1078
 为复制站点创建, 1078
 从 Interactive SQL, 86
 从 Sybase Central, 86
 从实用程序, 123
 使用 -ti 服务器选项删除, 212
 使用 LDAP, 138
 使用 login_procedure 选项设置最大数, 513
 使用数据源, 119
 关于, 79
 到数据库, 79
 到本地数据库, 117
 启动未连接的数据库, 54
 启动本地服务器, 117
 启用数据库功能, 544
 字符集, 386
 安全功能, 1005
 定义, 79
 实用程序数据库, 29
 审计, 1013
 属性, 566
 嵌入式数据库, 118
 性能, 129
 按字母顺序排序的属性列表, 566
 数据库连接方案, 116
 数据库镜像, 896
 最大临时文件空间, 552
 本地数据库, 116
 术语定义, 1105
 权限, 430
 查找服务器, 126
 概述, 79
 活动, 212
 测试嵌入式 SQL 的性能, 130
 疑难解答, 124
 登录策略, 416
 简单, 116
 编程接口, 83
 缺省参数, 122
 网络, 120
 自动启动数据库, 118

- 自动启动服务器, 126
- 详细信息, 124
- 连接到 Windows Mobile 上的数据库, 318
- 连接助手, 86
- 连接窗口概述, 86
- 通过 SQL Anywhere 控制台实用程序, 86
- 问题, 124
- 防火墙, 136
- 限制, 182
- 限制临时文件空间, 552
- 限制临时空间, 522
- 集成登录, 1006
- 验证, 491
- 验证 SQL Anywhere OEM 版, 72
- 连接 ID
 - 关于, 79
 - 术语定义, 1105
- 连接 Windows Mobile 设备
 - 关于, 318
- 连接参数
 - (参见 协议选项)
 - dbisqlc 实用程序, 720
 - Delphi, 714
 - DescribeCursor, 714
 - Description, 714
 - Driver, 714
 - GetTypeInfoChar, 714
 - InitString, 714
 - Interactive SQL [dbisql] 实用程序, 740
 - IsolationLevel, 714
 - 校验 [dbvalid] 实用程序, 810
 - KeysInSQLStatistics, 714
 - LazyAutocommit, 714
 - ODBC 数据源, 714
 - ping [dbping] 实用程序, 757
 - PreFetchOnOpen, 714
 - PreventNotCapable, 714
 - SQL Anywhere, 248
 - SQL Anywhere 控制台 [dbconsole] 实用程序, 780
 - SQL Anywhere 脚本执行 [dbrunsql] 实用程序, 761
 - SuppressWarnings, 714
 - TranslationDLL, 714
 - TranslationName, 714
 - TranslationOption, 714
 - 优先级, 80
 - 位置, 125
 - 使用缺省参数, 122
 - 信息 [dbinfo] 实用程序, 728
 - 停止 [dbstop] 实用程序, 783
 - 关于, 247
 - 冲突, 81
 - 区分大小写, 248
 - 升级 [dbupgrad] 实用程序, 808
 - 卸载 [dbunload] 实用程序, 795
 - 和连接字符串, 81
 - 备份 [dbbackup] 实用程序, 698
 - 嵌入式数据库, 81
 - 布尔值, 248
 - 建立连接, 83
 - 按字母顺序排序的列表, 247
 - 提示, 81
 - 数据源, 90
 - 数据源 [dbdsn] 实用程序, 710
 - 概述, 248
 - 空值, 81
 - 简介, 80
 - 翻译日志文件 [dbtran] 实用程序, 753
 - 设置, 81
- 连接窗口
 - 概述, 86
 - 访问, 87
 - 连接助手, 86
- 连接到实用程序数据库
 - 关于, 29
- 连接到数据库
 - Windows Mobile, 318
 - 关于, 79
- 连接的用户
 - 管理, 442
- 连接方案
 - 概述, 116
- 连接类型
 - 术语定义, 1105
- 连接配置
 - 术语定义, 1105
- 连接配置文件
 - 关于, 88
 - 创建, 88
 - 导入, 89
 - 导出, 89
 - 编辑, 89
 - 自动连接, 89

- 连接启动的同步
 - 术语定义, 1105
 - 连接属性
 - 区分大小写, 566
 - 报告, 757
 - 按字母顺序排序的列表, 566
 - 连接数据库
 - 关于, 79
 - 连接条件
 - 术语定义, 1105
 - 连接助手 (见 连接助手)
 - 关于, 86
 - 连接字符串
 - 重复参数的优先级, 80
 - 关于, 81
 - 字符集, 386
 - 按字母顺序排列的连接参数列表, 247
 - 空格, 80
 - 空连接参数, 81
 - 简介, 80
 - 表示, 80
 - 链接事务模式
 - chained 选项, 488
 - 列
 - 加密, 1024
 - 在 Interactive SQL 中查找, 648
 - 权限, 434
 - 限制, 614
 - 列表
 - 监控数据库选项, 465
 - 列名
 - 国际方面, 388
 - 列权限
 - 设置, 434
 - 临时 dbspace
 - 关于, 13
 - 权限, 24
 - 临时表
 - 术语定义, 1105
 - 限制, 614
 - 临时空间
 - 限制, 522
 - 临时文件
 - temp_space_limit_check 选项, 552
 - Unix 共享内存连接, 357
 - Windows Mobile 上的位置, 374
 - 使用 -dt 服务器选项指定位置, 170
 - 使用 SATMP 环境变量指定位置, 357
 - 使用 TEMP 环境变量指定位置, 366
 - 使用 TEMPDIR 环境变量指定位置, 366
 - 使用 TMP 环境变量指定位置, 366
 - 安全, 357
 - 连接使用的最大空间, 552
 - 限制, 614
 - 临时选项
 - Kerberos 登录安全, 115
 - 作用域和持续时间, 463
 - 设置, 462
 - 集成登录安全, 115
 - 零填充
 - 使用 date_format 选项控制, 497
 - 使用 timestamp_format 选项进行控制, 555
 - 另存为 ODBC 数据源
 - 关于, 91
 - 路由器
 - 广播, 291
 - 轮询
 - 术语定义, 1106
 - 设置频率, 65
 - 逻辑索引
 - 术语定义, 1106
 - 浏览计划查看器
 - 关于, 651
- ## M
- MachineName 属性
 - SQL Anywhere SNMP Extension Agent OID, 975
 - 服务器属性说明, 588
 - Mac OS X
 - dbconsole 实用程序硬件要求, 688
 - DYLD_LIBRARY_PATH 环境变量, 346
 - Interactive SQL 硬件要求, 636
 - IPv6 支持, 135
 - Sybase Central 硬件要求, 621
 - 创建 ODBC 数据源, 93
 - 发起文件, 344
 - 启动 dbconsole, 688
 - 启动 Interactive SQL, 638
 - 启动 Sybase Central, 621
 - 查看服务器消息, 218
 - 设置环境变量, 344
 - 连接到示例数据库, 117
 - MAGIC
 - user_estimates 选项, 559

- MainHeapBytes 属性
SQL Anywhere SNMP Extension Agent OID, 972
服务器属性说明, 588
- MainHeapPages 属性
SQL Anywhere SNMP Extension Agent OID, 972
服务器属性说明, 588
- MapPhysicalMemoryEng 属性
SQL Anywhere SNMP Extension Agent OID, 972
服务器属性说明, 588
- materialized_view_optimization 选项
SQL Anywhere SNMP Extension Agent OID, 987
说明, 515
连接属性说明, 566
- max_client_statements_cached 选项
SQL Anywhere SNMP Extension Agent OID, 987
说明, 516
连接属性说明, 566
- max_cursor_count 选项
SQL Anywhere SNMP Extension Agent OID, 987
说明, 517
连接属性说明, 566
- max_hash_size 选项
SQL Anywhere SNMP Extension Agent OID, 987
连接属性说明, 566
- max_plans_cached 选项
SQL Anywhere SNMP Extension Agent OID, 987
说明, 518
连接属性说明, 566
- max_priority 选项
SQL Anywhere SNMP Extension Agent OID, 987
说明, 518
连接属性说明, 566
- max_query_tasks 选项
SQL Anywhere SNMP Extension Agent OID, 987
说明, 519
连接属性说明, 566
- max_recursive_iterations 选项
SQL Anywhere SNMP Extension Agent OID, 987
说明, 520
连接属性说明, 566
- max_statement_count 选项
SQL Anywhere SNMP Extension Agent OID, 987
说明, 520
连接属性说明, 566
- max_temp_space 选项
SQL Anywhere SNMP Extension Agent OID, 987
说明, 522
连接属性说明, 566
- MaxCacheSize 属性
SQL Anywhere SNMP Extension Agent OID, 972
服务器属性说明, 588
- MaxConnections 属性
SQL Anywhere SNMP Extension Agent OID, 975
服务器属性说明, 588
- MaxConnections 协议选项
说明, 300
- MAXCONN 协议选项
说明, 300
- MaxEventType 属性
服务器属性说明, 588
- MaxEventType 属性
SQL Anywhere SNMP Extension Agent OID, 975
- MaxIO 属性
SQL Anywhere SNMP Extension Agent OID, 980
数据库属性说明, 601
- MaxMessage 属性
SQL Anywhere SNMP Extension Agent OID, 975
服务器属性说明, 588
- MaxRead 属性
SQL Anywhere SNMP Extension Agent OID, 980
数据库属性说明, 601
- MaxRemoteCapability 属性
SQL Anywhere SNMP Extension Agent OID, 975
服务器属性说明, 588
- MaxRequestSize 协议选项
说明, 301
- MAXSIZE 协议选项
说明, 301
- MaxWrite 属性
SQL Anywhere SNMP Extension Agent OID, 980
数据库属性说明, 601
- MessageCategoryLimit 属性
SQL Anywhere SNMP Extension Agent OID, 975
服务器属性说明, 588
- MessageReceived 属性
连接属性说明, 566
- MessageText 属性
SQL Anywhere SNMP Extension Agent OID, 975
服务器属性说明, 588
- MessageTime 属性
SQL Anywhere SNMP Extension Agent OID, 975
服务器属性说明, 588
- MessageWindowSize 属性
SQL Anywhere SNMP Extension Agent OID, 975

-
- 服务器属性说明, 588
 - Message 属性
 - SQL Anywhere SNMP Extension Agent OID, 975
 - 服务器属性说明, 588
 - MESSAGE 语句
 - 设置 debug_messages 选项, 498
 - ME 协议选项
 - 使用 IPv6 地址, 135
 - 说明, 301
 - MIB
 - (参见 管理信息库)
 - SQL Anywhere SNMP Extension Agent 支持, 957
 - 定义, 957
 - Microsoft Access
 - TIMESTAMP 比较, 500
 - MIME
 - 替代字符集编码标签, 404
 - min_password_length 选项
 - SQL Anywhere SNMP Extension Agent OID, 987
 - 提高口令安全性, 1007
 - 说明, 522
 - 连接属性说明, 566
 - MinCacheSize 属性
 - SQL Anywhere SNMP Extension Agent OID, 972
 - 服务器属性说明, 588
 - MirrorFailover 系统事件
 - 使用, 901
 - 说明, 869
 - MirrorMode 属性
 - SQL Anywhere SNMP Extension Agent OID, 984
 - 数据库属性说明, 601
 - 确定数据库镜像同步模式, 883
 - MirrorServerDisconnect 系统事件
 - 使用, 901
 - 说明, 869
 - MirrorServerName 参数
 - 使用, 901
 - MirrorState 属性
 - SQL Anywhere SNMP Extension Agent OID, 984
 - 数据库属性说明, 601
 - MIT Kerberos 客户端
 - Unix 支持, 107
 - Windows 支持, 107
 - mlasinst 实用程序
 - 在 Vista 上可能需要权限提升, 38
 - mlsrv11
 - 临时文件位置, 357
 - 启动支持传送层安全, 1047
 - 授权, 766
 - MobiLink
 - 数据库选项, 475
 - 术语定义, 1106
 - MobiLink 传送层安全
 - 关于, 1031
 - MobiLink 服务器
 - 术语定义, 1106
 - MobiLink 监控器
 - 术语定义, 1106
 - MobiLink 客户端
 - 数据库选项, 475
 - 术语定义, 1107
 - MobiLink 实用程序
 - MobiLink 证书创建 [createcert], 705
 - MobiLink 证书查看器 [viewcert] 实用程序, 708
 - MobiLink 同步
 - 备份, 847
 - 设置 default_timestamp_increment, 500
 - 设置 truncate_timestamp_values, 556
 - MobiLink 系统表
 - 术语定义, 1107
 - MobiLink 用户
 - 术语定义, 1107
 - MobiLink 证书查看器实用程序 [viewcert]
 - 语法, 708
 - MobiLink 证书创建实用程序 [createcert]
 - 语法, 705
 - MSDASQL OLE DB 提供程序
 - 关于, 97
 - MultiByteCharSet 属性
 - SQL Anywhere SNMP Extension Agent OID, 984
 - 数据库属性说明, 601
 - MultiPacketsReceived 属性
 - SQL Anywhere SNMP Extension Agent OID, 972
 - 服务器属性说明, 588
 - MultiPacketsSent 属性
 - SQL Anywhere SNMP Extension Agent OID, 972
 - 服务器属性说明, 588
 - MultiPageAllocs 属性
 - SQL Anywhere SNMP Extension Agent OID, 972
 - 服务器属性说明, 588
 - MultiProgrammingLevel 属性
 - SQL Anywhere SNMP Extension Agent OID, 975
 - 服务器属性说明, 588
 - MyIP 协议选项
-

- 使用 IPv6 地址, 135
- 说明, 301
- 每存取设备授权
 - 关于, 766
- 密钥对生成器实用程序 [createkey]
 - 语法, 744
- 密钥分发中心
 - 使用 Kerberos 验证, 108
- 名称
 - 数据库, 239
- 命令
 - 在 Interactive SQL 中重新调用, 645
 - 在 Interactive SQL 中中断, 647
 - 在 Interactive SQL 中停止, 647
 - 在 Interactive SQL 中取消, 647
 - 在 Interactive SQL 中执行, 641
 - 在 Interactive SQL 中编辑, 645
 - 在 Interactive SQL 中记录, 647
- 命令 shell
 - 大括号, xv
 - 引号, xv
 - 括号, xv
 - 环境变量, xv
 - 约定, xv
- 命令参数文件
 - 关于, 695
- 命令分隔符
 - dbisqlc 实用程序, 720
 - Interactive SQL [dbisql] 实用程序, 740
- 命令回写
 - echo 选项, 672
- 命令历史记录窗口
 - 在 Interactive SQL 中重新调用命令, 645
 - 在 Interactive SQL 中使用, 645
- 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
 - 使 Interactive SQL 成为缺省编辑器, 643
 - 术语定义, 1106
- 命令行
 - 使用配置文件, 695
 - 区分大小写, 40
 - 启动服务器, 40
 - 在配置文件中, 156
 - 常用选项, 43
 - 数据库服务器, 148
- 命令行实用程序
 - 重建语法, 760
 - createcert 语法, 705
 - createkey 语法, 744
 - dbisqlc 语法, 720
 - Interactive SQL [dbisql] 语法, 740
 - 校验 [dbvalid] 语法, 810
 - Linux 服务 [dbsvc] 语法, 769
 - MobiLink 证书创建 [createcert] 语法, 705
 - ping [dbping] 语法, 757
 - SQL Anywhere 控制台 [dbconsole] 实用程序, 780
 - SQL Anywhere 脚本执行 [dbrunsql] 语法, 761
 - viewcert 语法, 708
 - Windows 服务 [dbsvc] 语法, 773
 - 事务日志 [dblog] 语法, 792
 - 信息 [dbinfo] 语法, 728
 - 停止服务器 [dbstop] 语法, 783
 - 初始化 [dbinit] 语法, 729
 - 升级 [dbupgrad] 语法, 808
 - 卸载 [dbunload] 语法, 795
 - 后台启动服务器 [dbspawn] 语法, 781
 - 备份 [dbbackup] 语法, 698
 - 广播转发器 [dbns11] 语法, 703
 - 支持 [dbsupport] 语法, 785
 - 数据源 [dbdsn] 语法, 710
 - 文件隐藏 [dbfhide] 语法, 724
 - 日志传送管理器 [dbltn] 语法, 748
 - 服务器授权 [dblic] 语法, 766
 - 服务器枚举 [dblocate] 语法, 763
 - 消除 [dberase] 语法, 722
 - 直方图 [dbhist] 语法, 726
 - 翻译日志文件 [dbtran] 语法, 753
 - 语言选择 [dblang] 语法, 745
- 命令行选项
 - Interactive SQL [dbisql] 实用程序, 740
 - 校验 [dbvalid] 实用程序, 810
 - ping [dbping] 实用程序, 757
 - 事务日志 [dblog] 实用程序, 792
 - 信息 [dbinfo] 实用程序, 728
 - 初始化 [dbinit] 实用程序, 729
 - 升级 [dbupgrad] 实用程序, 808
 - 卸载 [dbunload] 实用程序, 795
 - 备份 [dbbackup] 实用程序, 698

数据源 [dbdsn] 实用程序, 710
 消除 [dberase] 实用程序, 722
 直方图 [dbhist] 实用程序, 726
 翻译日志文件 [dbtran] 实用程序, 753
 模式
 SQL Anywhere 插件, 630
 卸载定义, 795
 数据库镜像中的同步, 883
 术语定义, 1107
 目录访问服务器
 Windows Mobile 上不支持, 336
 目录结构
 SQL Anywhere, 368

N

Name 属性
 SQL Anywhere SNMP Extension Agent OID, 975, 984
 数据库属性说明, 601
 服务器属性说明, 588
 连接属性说明, 566
NAS
 存储数据库文件, 14
NativeProcessorArchitecture 属性
 SQL Anywhere SNMP Extension Agent OID, 975
 服务器属性说明, 588
nchar_charset
 关于, 386
nchar_charset 别名
 关于, 404
NcharCharSet 属性
 SQL Anywhere SNMP Extension Agent OID, 984
 数据库属性说明, 601
 连接属性说明, 566
NcharCollation 属性
 SQL Anywhere SNMP Extension Agent OID, 984
 数据库属性说明, 601
NCHAR 归类
 关于, 394
NCHAR 数据类型
 归类序列, 729
NDS
 文件名, 42
nearest_century 选项
 SQL Anywhere SNMP Extension Agent OID, 987
 Transact-SQL 兼容性, 473
 说明, 523
 连接属性说明, 566
net.cfg 文件
 客户端/服务器通信故障排除, 144
NewPassword 连接参数
 实现口令到期功能, 1007
 说明, 274
NEWPWD 连接参数
 实现口令到期功能, 1007
 说明, 274
NextScheduleTime 属性
 SQL Anywhere SNMP Extension Agent OID, 984
 数据库属性说明, 601
NIST
 FIPS 证书, 1033
NodeAddress 属性
 连接属性说明, 566
non_keywords 选项
 SQL Anywhere SNMP Extension Agent OID, 987
 Transact-SQL 兼容性, 473
 说明, 523
 连接属性说明, 566
NULL
 ANSI 行为, 484
 nulls 选项, 679
 Transact-SQL 行为, 484
nulls 选项
 Interactive SQL 设置, 667
 说明, 679
Number 属性
 连接属性说明, 566
NumLogicalProcessorsUsed 属性
 SQL Anywhere SNMP Extension Agent OID, 975
 服务器属性说明, 588
NumLogicalProcessors 属性
 SQL Anywhere SNMP Extension Agent OID, 975
 服务器属性说明, 588
NumPhysicalProcessorsUsed 属性
 SQL Anywhere SNMP Extension Agent OID, 975
 服务器属性说明, 588
NumPhysicalProcessors 属性
 SQL Anywhere SNMP Extension Agent OID, 975
 服务器属性说明, 588
内部
 调度, 873
 事件, 873
 事件处理, 873
 事件处理程序, 874

- 内部版本号
 - SQL Anywhere, 220
- 内部机制
 - 备份, 852
- 内部卸载
 - 使用, 805
- 内存
 - 设置初始高速缓存大小, 158
 - 设置最大高速缓存大小, 161
 - 设置最小高速缓存大小, 162
 - 设置静态高速缓存大小, 159
 - 连接限制, 457
 - 限制 AWE 高速缓存大小, 163
- 内存卡
 - Windows Mobile, 310
- 内存模式
 - 配置, 188
- 内核
 - 指定数据库服务器使用的数量, 186
- 内连接
 - 术语定义, 1107
- O**
- ODBC
 - Delphi, 524
 - odbc_describe_binary_as_varbinary 选项, 524
 - odbc_distinguish_char_and_varchar 选项, 525
 - Unix 支持, 95
 - Unix 的初始化文件, 95
 - Windows Mobile 上的受限制功能, 336
 - 使用 [连接] 窗口创建, 91
 - 使用 dbdsn 实用程序创建, 92
 - 使用 ODBC 管理器创建, 91
 - 在 Windows Mobile 上使用数据源, 94
 - 数据源, 90
 - 数据源连接参数, 714
 - 术语定义, 1107
 - 疑难解答, 757
 - 管理器, 91
 - 连接, 83
 - 连接参数, 248
 - 驱动程序位置, 124
 - odbc_describe_binary_as_varbinary 选项
 - SQL Anywhere SNMP Extension Agent OID, 987
 - 说明, 524
 - 连接属性说明, 566
 - odbc_distinguish_char_and_varchar 选项
 - SQL Anywhere SNMP Extension Agent OID, 987
 - 说明, 525
 - 连接属性说明, 566
 - ODBC_INI 环境变量
 - 定位系统信息文件, 95
 - 说明, 350
 - odbc.ini 文件
 - 关于, 95
 - ODBCHOME 环境变量
 - 说明, 349
 - ODBCINI 环境变量
 - 定位系统信息文件, 95
 - 说明, 350
 - ODBC INI 文件
 - 关于, 95
 - ODBC 管理器
 - 使用, 91
 - 术语定义, 1107
 - ODBC 连接参数
 - Delphi, 714
 - DescribeCursor, 714
 - Description, 714
 - Driver, 714
 - GetTypeInfoChar, 714
 - InitString, 714
 - IsolationLevel, 714
 - KeysInSQLStatistics, 714
 - LazyAutocommit, 714
 - PrefetchOnOpen, 714
 - PreventNotCapable, 714
 - SuppressWarnings, 714
 - TranslationDLL, 714
 - TranslationName, 714
 - TranslationOption, 714
 - ODBC 驱动程序
 - 线程和非线程版本, 93
 - 设置, 93
 - ODBC 示例
 - 使用, 316
 - ODBC 数据源
 - Unix, 95
 - 为 Windows Mobile 创建, 319
 - 使用 [另存为 ODBC 数据源] 创建, 91
 - 使用 dbdsn 创建, 710
 - 使用 ODBC 管理器创建, 91
 - 关于, 90
 - 在 Mac OS X 上创建, 93

- 术语定义, 1107
- 生成, 91
- 配置, 91
- oem_string 选项
 - SQL Anywhere SNMP Extension Agent OID, 987
 - 说明, 525
 - 连接属性说明, 566
- OEM 版
 - 关于, 70
- OEM 代码页
 - 关于, 386
- OID
 - (参见 对象标识符)
 - RDBMS MIB, 995
 - SQL Anywhere MIB, 969
 - 关于, 957
 - 定义, 957
 - 数据库属性, 983
 - 数据库统计信息, 980
 - 数据库选项, 987
 - 服务器属性, 975
 - 服务器统计信息, 972
- OLAP
 - optimization_workload 选项, 530
- OLE DB
 - SAOLEDB 提供程序, 97
 - 提供程序, 97
 - 连接, 97
- OmniConnect 支持
 - 关于, 1058
- OmniIdentifier 属性
 - SQL Anywhere SNMP Extension Agent OID, 975
 - 服务器属性说明, 588
- on_charset_conversion_failure 选项
 - SQL Anywhere SNMP Extension Agent OID, 987
 - 说明, 527
 - 连接属性说明, 566
- on_error 选项
 - Interactive SQL 设置, 667
 - 说明, 679
- on_tsq1_error 选项
 - ASE 兼容性, 473
 - Open Client, 1067
 - SQL Anywhere SNMP Extension Agent OID, 987
 - Transact-SQL 兼容性, 473
 - 说明, 528
 - 连接属性说明, 566
- ON EXCEPTION RESUME 子句
 - on_tsq1_error 选项, 528
- Open Client
 - ASE 兼容性选项, 473
 - Kerberos 验证, 110
 - Windows Mobile 上不支持, 336
 - 接口, 1058
 - 标识符的最大长度, 1083
 - 选项, 1067
 - 配置, 1062
- Open Server
 - 重命名服务器条目, 1066
 - 体系结构, 1058
 - 删除服务器条目, 1066
 - 启动, 1060
 - 地址, 1064
 - 添加, 1062
 - 系统要求, 1060
 - 连接, 1075
 - 配置用于 JDBC 的服务器, 1066
- OPENSTRING 子句
 - 查询文件所需的权限, 426
- optimistic_wait_for_commit 选项
 - SQL Anywhere SNMP Extension Agent OID, 987
- optimization_goal 选项
 - SQL Anywhere SNMP Extension Agent OID, 987
 - 说明, 528
 - 连接属性说明, 566
- optimization_level 选项
 - SQL Anywhere SNMP Extension Agent OID, 987
 - 说明, 529
 - 连接属性说明, 566
- optimization_workload 选项
 - SQL Anywhere SNMP Extension Agent OID, 987
 - 说明, 530
 - 连接属性说明, 566
- OptionWatchAction 属性
 - SQL Anywhere SNMP Extension Agent OID, 984
 - 关于, 465
 - 数据库属性说明, 601
 - 服务器属性说明, 588
- OptionWatchList 属性
 - SQL Anywhere SNMP Extension Agent OID, 984
 - 关于, 465
 - 数据库属性说明, 601
 - 服务器属性说明, 588
- Oracle 驱动程序

- 创建数据源, 711
- os_charset 别名
 - 关于, 404
- OSUser 属性
 - 连接属性说明, 566
- output_format 选项
 - Interactive SQL 设置, 667
 - 说明, 680
- output_length 选项
 - Interactive SQL 设置, 667
 - 说明, 681
- output_nulls 选项
 - Interactive SQL 设置, 667
 - 说明, 681
- Override-Magic
 - user_estimates 选项, 559
- P**
- PacketSize 属性
 - 连接属性说明, 566
- PacketsReceivedUncomp 属性
 - SQL Anywhere SNMP Extension Agent OID, 972
 - 服务器属性说明, 588
 - 连接属性说明, 566
- PacketsReceived 属性
 - SQL Anywhere SNMP Extension Agent OID, 972
 - 服务器属性说明, 588
 - 连接属性说明, 566
- PacketsSentUncomp 属性
 - SQL Anywhere SNMP Extension Agent OID, 972
 - 服务器属性说明, 588
 - 连接属性说明, 566
- PacketsSent 属性
 - SQL Anywhere SNMP Extension Agent OID, 972
 - 服务器属性说明, 588
 - 连接属性说明, 566
- PageRelocations 属性
 - SQL Anywhere SNMP Extension Agent OID, 980
 - 数据库属性说明, 601
- PageSize 属性
 - SQL Anywhere SNMP Extension Agent OID, 975, 984
 - 数据库属性说明, 601
 - 服务器属性说明, 588
- Palm HotSync 管道安装程序实用程序
 - 在 Vista 上可能需要权限提升, 38
- PartnerState 属性
 - SQL Anywhere SNMP Extension Agent OID, 984
 - 数据库属性说明, 601
- Password 连接参数
 - 说明, 276
- PATH 环境变量
 - 说明, 351
- PBUF 连接参数
 - 说明, 277
- PDB
 - 术语定义, 1107
- PDF
 - 文档, xii
- PeakCacheSize 属性
 - SQL Anywhere SNMP Extension Agent OID, 972
 - 服务器属性说明, 588
- PEM 编码的 PKI 对象
 - 查看, 708
- ping 实用程序 [dbping]
 - TCP/IP, 143
 - 使用, 130
 - 测试 Open Client, 1065
 - 测试网络, 68
 - 语法, 757
 - 退出代码, 759
- pinned_cursor_percent_of_cache 选项
 - SQL Anywhere SNMP Extension Agent OID, 987
 - 说明, 531
 - 连接属性说明, 566
- PKI 对象
 - 查看, 708
- PlatformVer 属性
 - SQL Anywhere SNMP Extension Agent OID, 975
 - 服务器属性说明, 588
- Platform 属性
 - SQL Anywhere SNMP Extension Agent OID, 975
 - 服务器属性说明, 588
- PORT 协议选项
 - 将 SQL Anywhere 用作 Open Server, 1061
 - 说明, 303
- post_login_procedure 选项
 - SQL Anywhere SNMP Extension Agent OID, 987
 - 实现口令到期功能, 1007
 - 说明, 532
 - 连接属性说明, 566
- PowerBuilder DataWindow
 - 查询性能, 528
- PowerDesigner

-
- 术语定义, 1107
 - PowerJ
 - 术语定义, 1108
 - precision 选项
 - SQL Anywhere SNMP Extension Agent OID, 987
 - 说明, 533
 - 连接属性说明, 566
 - PrefetchBuffer 连接参数
 - 说明, 277
 - PrefetchOnOpen 连接参数
 - ODBC 连接参数说明, 714
 - PreFetchOnOpen 连接参数
 - 说明, 277
 - PrefetchRows 连接参数
 - 说明, 278
 - prefetch 选项
 - SQL Anywhere SNMP Extension Agent OID, 987
 - 说明, 534
 - 连接属性说明, 566
 - Prepares 属性
 - 数据库属性说明, 601
 - 连接属性说明, 566
 - PrepStmt 属性
 - 连接属性说明, 566
 - preserve_source_format 选项
 - SQL Anywhere SNMP Extension Agent OID, 987
 - 说明, 534
 - 连接属性说明, 566
 - prevent_article_pkey_update 选项
 - SQL Anywhere SNMP Extension Agent OID, 987
 - 说明, 535
 - 连接属性说明, 566
 - PreventNotCapable 连接参数
 - ODBC 连接参数说明, 714
 - priority 选项
 - SQL Anywhere SNMP Extension Agent OID, 987
 - 说明, 535
 - 连接属性说明, 566
 - ProcedurePages 属性
 - SQL Anywhere SNMP Extension Agent OID, 980
 - 数据库属性说明, 601
 - ProcedureProfiling 属性
 - SQL Anywhere SNMP Extension Agent OID, 984
 - 数据库属性说明, 601
 - ProcessCPUSystem 属性
 - SQL Anywhere SNMP Extension Agent OID, 975
 - 服务器属性说明, 588
 - ProcessCPUUser 属性
 - SQL Anywhere SNMP Extension Agent OID, 975
 - 服务器属性说明, 588
 - ProcessCPU 属性
 - SQL Anywhere SNMP Extension Agent OID, 975
 - 服务器属性说明, 588
 - ProcessorArchitecture 属性
 - SQL Anywhere SNMP Extension Agent OID, 975
 - 服务器属性说明, 588
 - ProductName 属性
 - SQL Anywhere SNMP Extension Agent OID, 975
 - 服务器属性说明, 588
 - ProductVersion 属性
 - SQL Anywhere SNMP Extension Agent OID, 975
 - 服务器属性说明, 588
 - ProfileFilterConn 属性
 - SQL Anywhere SNMP Extension Agent OID, 975
 - 服务器属性说明, 588
 - ProfileFilterUser 属性
 - SQL Anywhere SNMP Extension Agent OID, 975
 - 服务器属性说明, 588
 - PROFILE 特权
 - 关于, 425
 - 可继承的, 425
 - 授予, 433
 - PROPERTY 函数
 - 按字母顺序排序的数据库服务器属性列表, 588
 - PROWS 连接参数
 - 说明, 278
 - PUBLIC 选项
 - 关于, 463
 - 要求 DBA 权限, 463
 - PUBLIC 组
 - 关于, 448
 - PWD 连接参数
 - 说明, 276
 - 排序
 - 结果集, 657
 - 排序顺序
 - 定义, 384
 - 归类, 378
 - 配置
 - LTM, 1086
 - ODBC 数据源, 91
 - SQL Anywhere 控制台 [dbconsole] 实用程序, 688
 - sql.ini, 1062

- Windows Mobile 数据库, 321
 - 使用 dbconsole, 780
 - 复制服务器的 SQL Anywhere 数据库, 1081
 - 复制服务器的 SQL Anywhere 数据库用户, 1082
 - 接口文件, 1062
 - 文本完成, 684
- 配置 MobiLink 客户端以使用传送层安全
 - 关于, 1049
- 配置 SQL Anywhere 客户端以使用传送层安全
 - 关于, 1050
- 配置 UltraLite 客户端以使用传送层安全
 - 关于, 1052
- 配置数据库
 - Windows Mobile, 321
- 配置数据源
 - 使用 ODBC 管理器, 91
- 配置文件
 - (参见 @data 选项)
 - LTM, 1086
 - LTM 命令行, 748
 - LTM 的格式, 1087
 - 关于, 695
 - 创建 LTM, 1087
 - 日志传送管理器 [dbltm] 实用程序, 748
 - 条件分析, 696
 - 用 dbfhide 实现简单加密, 724
 - 用于 LTM, 749
 - 选项, 156
 - 隐藏, 724
- 批处理模式
 - LTM 的, 1087
- 批处理文件
 - 使用 dbspawn 启动数据库服务器, 155
 - 启动服务器, 781
- 批量操作
 - b 服务器选项, 157
- 批量装载
 - 选项, 46
- 票据授予票据
 - Kerberos, 110
- 七位字符
 - 关于, 385
- Q**
- QAnywhere
 - 术语定义, 1108
- QAnywhere 代理
 - 术语定义, 1108
- qualify_owners 选项
 - SQL Remote 复制选项, 476
 - 说明, 536
- qualify_table_owners 参数
 - LTM 配置文件, 750
- query_mem_timeout 选项
 - SQL Anywhere SNMP Extension Agent OID, 987
 - 说明, 536
 - 连接属性说明, 566
- query_plan_on_open 选项
 - Transact-SQL 兼容性, 473
- QueryBypassedCosted 属性
 - 数据库属性说明, 601
 - 连接属性说明, 566
- QueryBypassedHeuristic 属性
 - 数据库属性说明, 601
 - 连接属性说明, 566
- QueryBypassedOptimized 属性
 - 数据库属性说明, 601
 - 连接属性说明, 566
- QueryBypassed 属性
 - 数据库属性说明, 601
 - 连接属性说明, 566
- QueryCachedPlans 属性
 - 数据库属性说明, 601
 - 连接属性说明, 566
- QueryCachePages 属性
 - SQL Anywhere SNMP Extension Agent OID, 980
 - 数据库属性说明, 601
 - 连接属性说明, 566
- QueryDescribedBypass 属性
 - 数据库属性说明, 601
 - 连接属性说明, 566
- QueryDescribedOptimizer 属性
 - 数据库属性说明, 601
 - 连接属性说明, 566
- QueryHeapPages 属性
 - SQL Anywhere SNMP Extension Agent OID, 972
 - 服务器属性说明, 588
 - 连接属性说明, 566
- QueryJHToJNLOptUsed 属性
 - 数据库属性说明, 601
 - 连接属性说明, 566
- QueryLowMemoryStrategy 属性
 - SQL Anywhere SNMP Extension Agent OID, 980
 - 数据库属性说明, 601

-
- 连接属性说明, 566
 - QueryMemActiveCurr 属性
 - SQL Anywhere SNMP Extension Agent OID, 972
 - 连接属性说明, 566
 - QueryMemActiveEst 属性
 - SQL Anywhere SNMP Extension Agent OID, 972
 - 服务器属性说明, 588
 - QueryMemActiveMax 属性
 - SQL Anywhere SNMP Extension Agent OID, 975
 - 服务器属性说明, 588
 - QueryMemExtraAvail 属性
 - SQL Anywhere SNMP Extension Agent OID, 972
 - 连接属性说明, 566
 - QueryMemGrantBaseMI 属性
 - SQL Anywhere SNMP Extension Agent OID, 975
 - 服务器属性说明, 588
 - QueryMemGrantBase 属性
 - SQL Anywhere SNMP Extension Agent OID, 975
 - 服务器属性说明, 588
 - QueryMemGrantExtra 属性
 - SQL Anywhere SNMP Extension Agent OID, 975
 - 服务器属性说明, 588
 - QueryMemGrantFailed 属性
 - SQL Anywhere SNMP Extension Agent OID, 972
 - 连接属性说明, 566
 - QueryMemGrantGranted 属性
 - SQL Anywhere SNMP Extension Agent OID, 972
 - 连接属性说明, 566
 - QueryMemGrantRequested 属性
 - SQL Anywhere SNMP Extension Agent OID, 972
 - 连接属性说明, 566
 - QueryMemGrantWaited 属性
 - SQL Anywhere SNMP Extension Agent OID, 972
 - 服务器属性说明, 588
 - 连接属性说明, 566
 - QueryMemGrantWaiting 属性
 - SQL Anywhere SNMP Extension Agent OID, 972
 - 连接属性说明, 566
 - QueryMemPages 属性
 - SQL Anywhere SNMP Extension Agent OID, 975
 - 服务器属性说明, 588
 - QueryMemPercentOfCache 属性
 - SQL Anywhere SNMP Extension Agent OID, 975
 - 服务器属性说明, 588
 - QueryMemWaited 属性
 - 连接属性说明, 566
 - QueryOpened 属性
 - 数据库属性说明, 601
 - 连接属性说明, 566
 - QueryOptimized 属性
 - 数据库属性说明, 601
 - 连接属性说明, 566
 - QueryReused 属性
 - 数据库属性说明, 601
 - 连接属性说明, 566
 - QueryRowsBufferFetch 属性
 - 数据库属性说明, 601
 - 连接属性说明, 566
 - QueryRowsMaterialized 属性
 - SQL Anywhere SNMP Extension Agent OID, 980
 - 数据库属性说明, 601
 - 连接属性说明, 566
 - 区分重音
 - 使用法语规则, 729
 - 数据库, 729
 - QuittingTime 属性
 - SQL Anywhere SNMP Extension Agent OID, 975
 - 服务器属性说明, 588
 - quote_all_identifiers 选项
 - SQL Remote 复制选项, 476
 - 说明, 537
 - quoted_identifier 选项
 - Open Client, 1067
 - SQL Anywhere SNMP Extension Agent OID, 987
 - Transact-SQL 兼容性, 473
 - 说明, 537
 - 连接属性说明, 566
 - 企业根证书
 - 传送层安全, 1036, 1037, 1039
 - 创建, 707
 - 启动 MobiLink 服务器
 - 传送层安全, 1047
 - 启动数据库
 - 关于, 54
 - 无连接, 54
 - 启用
 - 表加密, 1027
 - 迁移数据库向导
 - Windows Mobile 上不支持, 339
 - 签出
 - Interactive SQL 中的文件, 661
 - 签名
 - ECC 和 RSA 证书, 705
 - Vista 的可执行文件, 39

- 获取经过验证的应用程序, 70
- 签名证书
 - 在传送层安全中创建, 1039
- 签入
 - Interactive SQL 中的文件, 662
- 前导空格
 - 在连接字符串内使用, 81
- 前滚日志
 - 关于, 14
- 嵌入式 SQL
 - 接口库, 124
 - 术语定义, 1108
 - 测试连接的性能, 130
 - 连接, 83
 - 连接性能, 130
- 嵌入式数据库
 - 启动, 118
 - 指定数据库服务器名称, 119
 - 连接, 118
 - 连接参数, 81
- 强制回应
 - 服务器, 757
- 清除
 - SQL 语句窗格, 640
- 情形
 - 数据库镜像系统中的故障转移, 902
- 请求
 - SQL Anywhere 中的线程, 47
- 请求级记录 (无请求记录)
- 请求记录
 - 将记录信息保存到文件中, 229
 - 数据库服务器选项, 230
 - 请求日志副本的数量, 228
 - 限制日志文件的大小, 232
- 请求日志
 - 使用, 229
 - 副本的数量, 228
 - 限制大小, 232
- 区分
 - 重音, 396
 - 大小写, 396
 - 标点符号, 396
- 区分大小写
 - dbinit 实用程序, 729
 - 不区分大小写的土耳其语数据库, 412
 - 初始化 [dbinit] 实用程序, 729
 - 命令行, 40
 - 土耳其语区分大小写的数据库, 411
 - 数据库名, 44
 - 数据库选项, 463
 - 服务器名, 44
 - 连接参数, 248
- 区域设置
 - 关于, 389
 - 字符集, 386, 391
 - 确定, 399
 - 设置, 400
 - 语言, 389
- 区域设置定义
 - 关于, 389
- 驱动程序
 - SQL Anywhere ODBC 驱动程序, 90
- 取消未提交的错误报告
 - 监控器, 949
- 权限
 - BACKUP 特权, 424
 - DBA 特权, 425
 - PROFILE 特权, 425
 - READCLIENTFILE 特权, 426
 - READFILE 特权, 426
 - REMOTE DBA 特权, 427
 - RESOURCE 特权, 427
 - VALIDATE 特权, 427
 - WITH GRANT OPTION, 437
 - WRITECLIENTFILE 特权, 427
 - 不支持负权限, 424
 - 临时文件, 357
 - 使用视图获得更高的安全性, 452
 - 关于, 428
 - 冲突, 456
 - 列表, 458
 - 单个, 430
 - 卸载数据, 181
 - 口令, 432
 - 安全功能, 1006
 - 授予 REMOTE, 439
 - 授予口令, 430
 - 授予权限, 437
 - 撤消, 439
 - 撤消 REMOTE, 439
 - 文件管理语句, 30
 - 模式, 1006
 - 管理, 415
 - 组, 429, 443

- 组成员资格, 444
- 继承, 422, 437, 443
- 表, 428, 434
- 装载数据, 181
- 视图, 428
- 触发器, 427, 438
- 设置表权限, 434
- 过程, 437
- 过程的设置, 437
- 连接, 430
- 选项, 45
- 针对视图授予, 435
- 集成登录权限, 99
- 全局临时表
 - 术语定义, 1108
- 全局签名证书
 - 传送层安全, 1039
- 全局证书
 - 作为服务器证书用于传送层安全, 1040
- 缺省
 - 连接参数, 122
- 缺省数据库服务器
 - 关于, 195
- 缺省字符集
 - about, 391
 - Unix, 391
 - Windows, 391

R

- RAISERROR 系统事件
 - 说明, 869
- RAISERROR 语句
 - continue_after_raiserror 选项, 492
 - on_tsq_error 选项, 528
- RAS
 - 拨号网络, 137
- RCVBUFSZ 协议选项
 - 说明, 302
- RDBMS
 - 术语定义, 1100
- rdbmsDbInfoTable
 - 说明, 995
- rdbmsDbLimitedResourceTable
 - 说明, 997
- rdbmsDbParamTable
 - 说明, 996
- rdbmsDbTable

- 说明, 995
- RDBMS MIB
 - 关于, 959
 - 表列表, 995
- RDBMS-MIB.mib 文件
 - 位置, 995
 - 关于, 956, 959
- rdbmsSrvInfoTable
 - 说明, 998
- rdbmsSrvLimitedResourceTable
 - 说明, 1000
- rdbmsSrvParamTable
 - 说明, 999
- rdbmsSrvTable
 - 说明, 997
- read_authdn 参数
 - LDAP, 139
- read_password 参数
 - LDAP, 139
- read_past_deleted 选项
 - SQL Anywhere SNMP Extension Agent OID, 987
 - 说明, 538
 - 连接属性说明, 566
- READCLIENTFILE 特权
 - 关于, 426
 - 可继承的, 426
 - 授予, 433
- READFILE 特权
 - 关于, 426
 - 可继承的, 426
 - 授予, 433
- ReadOnly 属性
 - SQL Anywhere SNMP Extension Agent OID, 984
 - 数据库属性说明, 601
- REATE DATABASE 语句
 - Windows Mobile 上的限制, 337
- ReceiveBufferSize 协议选项
 - 说明, 302
- ReceivingTracingFrom 属性
 - SQL Anywhere SNMP Extension Agent OID, 984
 - 数据库属性说明, 601
- recovery_time 选项
 - SQL Anywhere SNMP Extension Agent OID, 987
 - 使用, 853
 - 说明, 538
 - 连接属性说明, 566
- RecoveryUrgency 属性

- SQL Anywhere SNMP Extension Agent OID, 980
 - 数据库属性说明, 601
- RecursiveIterationsHash 属性
 - 数据库属性说明, 601
 - 连接属性说明, 566
- RecursiveIterationsNested 属性
 - 数据库属性说明, 601
 - 连接属性说明, 566
- RecursiveIterations 属性
 - 数据库属性说明, 601
 - 连接属性说明, 566
- RecursiveJNLMisses 属性
 - 数据库属性说明, 601
 - 连接属性说明, 566
- RecursiveJNLProbes 属性
 - 数据库属性说明, 601
 - 连接属性说明, 566
- REFERENCES 权限
 - 关于, 428
 - 授予, 434
- RelocatableHeapPages 属性
 - SQL Anywhere SNMP Extension Agent OID, 980
 - 数据库属性说明, 601
- RememberLastPlan 属性
 - SQL Anywhere SNMP Extension Agent OID, 975
 - 服务器属性说明, 588
- RememberLastStatement 属性
 - SQL Anywhere SNMP Extension Agent OID, 975
 - 服务器属性说明, 588
- remote_idle_timeout 选项
 - SQL Anywhere SNMP Extension Agent OID, 987
 - 说明, 539
 - 连接属性说明, 566
- RemoteCapability 属性
 - SQL Anywhere SNMP Extension Agent OID, 975
 - 服务器属性说明, 588
- REMOTE DBA 权限
 - 术语定义, 1120
 - 联机备份的要求, 817
- REMOTE DBA 特权
 - 关于, 427
- RemoteputWait 属性
 - SQL Anywhere SNMP Extension Agent OID, 972
 - 服务器属性说明, 588
- RemoteTrunc 属性
 - SQL Anywhere SNMP Extension Agent OID, 984
 - 数据库属性说明, 601
- REMOTE 权限
 - 授予和撤消, 439
- REMOVE JAVA 语句
 - Windows Mobile 上不支持, 337
- REORGANIZE TABLE 语句
 - Windows Mobile 上不支持, 337
- rep_func 参数
 - LTM 配置文件, 750
- replicate_all 选项
 - SQL Anywhere SNMP Extension Agent OID, 987
 - 复制代理选项, 477
 - 说明, 539
 - 连接属性说明, 566
- REPLICATE ON 子句
 - 配合 ALTER TABLE 语句使用, 1076
- replication_error_piece 选项
 - SQL Remote 复制选项, 476
 - 说明, 540
- replication_error 选项
 - SQL Remote 复制选项, 476
 - 说明, 539
- Replication Server
 - 日志传送管理器, 748
- ReqCountActive 属性
 - 连接属性说明, 566
- ReqCountBlockContention 属性
 - 连接属性说明, 566
- ReqCountBlockIO 属性
 - 连接属性说明, 566
- ReqCountBlockLock 属性
 - 连接属性说明, 566
- ReqCountUnscheduled 属性
 - 连接属性说明, 566
- ReqStatus 属性
 - 连接属性说明, 566
- ReqTimeActive 属性
 - 连接属性说明, 566
- ReqTimeBlockContention 属性
 - 连接属性说明, 566
- ReqTimeBlockIO 属性
 - 连接属性说明, 566
- ReqTimeBlockLock 属性
 - 连接属性说明, 566
- ReqTimeUnscheduled 属性
 - 连接属性说明, 566
- ReqType 属性
 - 连接属性说明, 566

-
- request_timeout 选项
 - SQL Anywhere SNMP Extension Agent OID, 987
 - 说明, 540
 - 连接属性说明, 566
 - RequestFilterConn 属性
 - SQL Anywhere SNMP Extension Agent OID, 975
 - 服务器属性说明, 588
 - RequestFilterDB 属性
 - SQL Anywhere SNMP Extension Agent OID, 975
 - 服务器属性说明, 588
 - RequestLogFile 属性
 - SQL Anywhere SNMP Extension Agent OID, 975
 - 服务器属性说明, 588
 - RequestLogging 属性
 - SQL Anywhere SNMP Extension Agent OID, 975
 - 服务器属性说明, 588
 - RequestLogMaxSize 属性
 - SQL Anywhere SNMP Extension Agent OID, 975
 - 服务器属性说明, 588
 - RequestLogNumFiles 属性
 - SQL Anywhere SNMP Extension Agent OID, 975
 - 服务器属性说明, 588
 - RequestsReceived 属性
 - SQL Anywhere SNMP Extension Agent OID, 972
 - 服务器属性说明, 588
 - 连接属性说明, 566
 - RequestTiming 属性
 - SQL Anywhere SNMP Extension Agent OID, 975
 - 服务器属性说明, 588
 - Req 属性
 - SQL Anywhere SNMP Extension Agent OID, 972
 - 服务器属性说明, 588
 - RESOURCE 特权
 - 不可继承, 427
 - 关于, 427
 - 授予, 433
 - RetryConnectionTimeout 连接参数
 - 说明, 279
 - 连接到镜像数据库, 897
 - RetryConnTO 连接参数
 - 说明, 279
 - 连接到镜像数据库, 897
 - return_date_time_as_string 选项
 - 说明, 541
 - 连接属性说明, 566
 - Rlbk 属性
 - 连接属性说明, 566
 - rollback_on_deadlock 选项
 - SQL Anywhere SNMP Extension Agent OID, 987
 - 说明, 542
 - 连接属性说明, 566
 - RollbackLogPages 属性
 - SQL Anywhere SNMP Extension Agent OID, 980
 - 数据库属性说明, 601
 - 连接属性说明, 566
 - ROLLBACK 语句
 - 日志, 855
 - 游标, 481
 - row_counts 选项
 - SQL Anywhere SNMP Extension Agent OID, 987
 - 说明, 542
 - 连接属性说明, 566
 - RS_pw 参数
 - LTM 配置文件, 750
 - 启动 LTM, 1079
 - RS_source_db 参数
 - LTM 配置文件, 750
 - 启动 LTM, 1079
 - RS_source_ds 参数
 - LTM 配置文件, 750
 - 启动 LTM, 1079
 - RS_user 参数
 - LTM 配置文件, 750
 - 启动 LTM, 1079
 - RSA
 - 支持, 1033
 - RSA 选项
 - dbeng11 -ec, 172
 - dbsrv11 -ec, 172
 - RSA 证书
 - 创建, 705
 - 查看, 708
 - rssetup.sql 脚本
 - 关于, 1081
 - 准备运行, 1081
 - 运行, 1082
 - RS 参数
 - LTM 配置文件, 750
 - 启动 LTM, 1079
 - 任务
 - 调度, 875
 - SQL Anywhere 中的线程, 47
 - 事件, 875
 - 任务列表
-

- 显示, 622
- 日期
 - date_order 选项, 497
 - nearest_century 选项, 523
- 日志查看器
 - 关于, 630
 - 打开, 630
- 日志传送管理器实用程序 [dbltm]
 - 使用, 1084
 - 关于, 1058
 - 标识符, 1083
 - 组件, 1071
 - 语法, 748
 - 退出代码, 749
- 日志文件
 - echo 选项, 672
 - 事务, 14
 - 事务日志 [dblog] 实用程序, 794
 - 事务日志镜像, 15
 - 回退, 855
 - 审计, 1017
 - 数据库服务器消息日志, 40
 - 数据库镜像, 901
 - 术语定义, 1108
 - 检查点日志, 17
- 软件
 - 授权数据库服务器, 766
 - 更新, 690
 - 版本, 220
- 软件版本
 - 数据库服务器, 220
 - 确定, 813
- 软件更新
 - 获取, 690
- 软件许可
 - 授权服务器, 766
- 瑞典语
 - UCA 归类, 398
- S**
- sa_config.csh 文件
 - 发起, 344
- sa_config.sh 文件
 - 发起, 344
- sa_conn_properties 系统过程
 - 按字母顺序排序的连接属性列表, 566
 - 用法, 464
- sa_db_properties 系统过程
 - 按字母顺序排序的数据库属性列表, 601
- sa_eng_properties 系统过程
 - 按字母顺序排序的数据库服务器属性列表, 588
- sa_monitor_connection_failed_event 事件
 - 监控器, 950
- sa_monitor_connection_failure 表
 - 监控器, 950
- sa_monitor_count_unsubmitted_crash_reports 函数
 - 监控器, 950
- sa_monitor_user
 - 关于, 950
- sa_server_option 系统过程
 - 监控数据库选项设置, 465
- SACA
 - UTF-8 字符集, 392
 - 关于, 392
 - 单字节字符集, 392
 - 多字节字符集, 392
- SACHARSET 环境变量
 - 指定字符集, 352
- SADatabase 代理
 - 测试, 909
 - 配置, 907
- saDbOptMetaDataTable
 - SQL Anywhere MIB, 972
- saDbPropMetaDataTable
 - SQL Anywhere MIB, 971
- saDbStatMetaDataTable
 - SQL Anywhere MIB, 971
- SADIAGDIR 环境变量
 - 指定诊断信息的位置, 353
- SALANG 环境变量
 - 指定语言, 355
- SALOGDIR 环境变量
 - 说明, 356
- samples-dir
 - 关于, 368
 - 文档用法, xiv
- SAN
 - 存储数据库文件, 14
- SAOLEDB
 - 连接 SQL Anywhere, 97
- SAServer 代理
 - 测试, 907
 - 配置, 906
- sasmp.ini 文件

SQL Anywhere SNMP Extension Agent 需要, 961
关于, 956

sasrv.ini 文件
排除服务器启动故障, 68
服务器信息, 129

saSrvPropMetaDataTable
SQL Anywhere MIB, 970

saSrvStatMetaDataTable
SQL Anywhere MIB, 970

SATMP 环境变量
Unix, 366
说明, 357

save_remote_passwords 选项
SQL Remote 选项, 543

scale 选项
SQL Anywhere SNMP Extension Agent OID, 987
说明, 543
连接属性说明, 566

scan_retry 参数
LTM 配置文件, 750
启动 LTM, 1079

scjview
关于, 621

scjview.exe
关于, 621
快速启动程序选项, 686

scripts 目录
关于, 368

search_timeout 参数
LDAP, 139

secure_feature_key 选项
SQL Anywhere SNMP Extension Agent OID, 987
说明, 544
连接属性说明, 566

SecureFeatures 属性
-sf 服务器选项, 207

SELECT 权限
关于, 428
授予, 434

SELinux 策略
使用 SQL Anywhere 策略, 1004

SendBufferSize 协议选项
说明, 303

SendFail 属性
SQL Anywhere SNMP Extension Agent OID, 972
服务器属性说明, 588

SendingTracingTo 属性
SQL Anywhere SNMP Extension Agent OID, 984
数据库属性说明, 601

ServerEdition 属性
服务器属性说明, 588

ServerIdle 系统事件
示例, 870
说明, 869

ServerName 连接参数
字符集, 80
嵌入式数据库, 119
说明, 280

ServerName 属性
SQL Anywhere SNMP Extension Agent OID, 975
服务器属性说明, 588

ServerNodeAddress 属性
连接属性说明, 566

ServerPort 属性
连接属性说明, 566

ServerPort 协议选项
将 SQL Anywhere 用作 Open Server, 1061
说明, 303

SessionCreateTime 属性
连接属性说明, 566

SessionID 属性
连接属性说明, 566

SessionLastTime 属性
连接属性说明, 566

SessionTimeout 属性
连接属性说明, 566

SET OPTION 语句
Interactive SQL 选项, 667
使用, 462

SET TEMPORARY OPTION 语句
使用, 462

SetupVSPackage.exe
在 Vista 上可能需要权限提升, 38

shell 模式
查看数据库服务器消息, 215
使用调度和事件自动完成任务
关于, 863

SHLIB_PATH 环境变量
说明, 359

数据库服务器消息日志
重命名和重新启动, 198

数据库校验
使用校验实用程序, 810
关于, 832, 858

- SMP
 - 处理器数量, 45, 185
- SnapshotCount 属性
 - SQL Anywhere SNMP Extension Agent OID, 980
 - 数据库属性说明, 601
 - 连接属性说明, 566
- SnapshotIsolationState 属性
 - SQL Anywhere SNMP Extension Agent OID, 984
 - 数据库属性说明, 601
- SNDBUFSZ 协议选项
 - 说明, 303
- SNMP
 - 代理, 957
 - 使用 SQL Anywhere SNMP Extension Agent, 955
 - 使用陷阱, 965
 - 关于, 957
 - 动态陷阱, 966
 - 安装, 961
 - 管理器, 957
 - 陷阱, 957
- SNMPv2-SMI.mib 文件
 - 关于, 956
- SNMPv2-TC.mib 文件
 - 关于, 956
- SNMP 服务
 - 重新启动, 961
- Solaris
 - IPv6 支持, 135
 - LD_LIBRARY_PATH 环境变量, 347
 - 在 shell 模式下查看服务器消息, 215
- sort_collation 选项
 - SQL Anywhere SNMP Extension Agent OID, 987
 - 说明, 544
 - 连接属性说明, 566
- SortMergePasses 属性
 - 数据库属性说明, 601
 - 连接属性说明, 566
- SortRowsMaterialized 属性
 - 数据库属性说明, 601
 - 连接属性说明, 566
- SortRunsWritten 属性
 - 数据库属性说明, 601
 - 连接属性说明, 566
- SortSortedRuns 属性
 - 数据库属性说明, 601
 - 连接属性说明, 566
- SortWorkTables 属性
 - 数据库属性说明, 601
 - 连接属性说明, 566
- sp_setreplicate 过程
 - 关于, 1086
- sp_setrepproc 过程
 - 关于, 1086
- SQL
 - Windows Mobile 上不支持的语句, 337
 - 术语定义, 1112
- SQL_database 参数
 - LTM 配置文件, 750
 - 启动 LTM, 1079
- sql_flagger_error_level 选项
 - SQL Anywhere SNMP Extension Agent OID, 987
 - Transact-SQL 兼容性, 473
 - 说明, 545
 - 连接属性说明, 566
- sql_flagger_warning_level 选项
 - SQL Anywhere SNMP Extension Agent OID, 987
 - Transact-SQL 兼容性, 473
 - 说明, 546
 - 连接属性说明, 566
- SQL_pw 参数
 - LTM 配置文件, 750
 - 启动 LTM, 1079
- SQL_server 参数
 - LTM 配置文件, 750
 - 启动 LTM, 1079
- SQL_user 参数
 - LTM 配置文件, 750
 - 启动 LTM, 1079
- SQL/2003 遵从性
 - SQL_FLAGGER_ERROR 选项, 545
 - 更新, 483
- sql.ini 文件
 - 关于, 1074
 - 配置, 1062
- SQLANY11 环境变量
 - 说明, 360
- SQLANYSAMP11 环境变量
 - 说明, 361
- SQL Anywhere
 - Windows Mobile 上不支持的功能, 336
 - 使用 SQL Anywhere, 309
 - 使用数据源连接, 119
 - 国际功能, 380
 - 在 Vista 上运行, 38

- 在 Windows Mobile 上使用, 309
- 安装 SQL Anywhere, 310
- 文档, xii
- 本地化版本, 378
- 术语定义, 1112
- 经过验证的应用程序, 70
- 设置为 Open Server, 1060
- 软件更新, 690
- 配置 Web 服务器以使用传送层安全, 1046
- 配置 Windows Mobile 数据库, 321
- 配置数据库服务器以使用传送层安全, 1042, 1043
- 需要的 ActiveSync 版本, 310
- SQL Anywhere for Windows
 - 使用 SQL Anywhere 服务器示例, 313
- SQL Anywhere MIB
 - Agent 表, 969
 - saDbOptMetaDataTable, 972
 - saDbPropMetaDataTable, 971
 - saDbStatMetaDataTable, 971
 - saMetaData 表, 969
 - saSrvPropMetaDataTable, 970
 - saSrvStatMetaDataTable, 970
 - 关于, 957
 - 数据库属性, 983
 - 数据库统计信息, 980
 - 数据库选项, 987
 - 服务器属性, 975
 - 服务器统计信息, 972
 - 表列表, 969
- SQL Anywhere MIB 参考
 - 概述, 969
- SQL Anywhere ODBC 驱动程序
 - 关于, 90
- SQL Anywhere OEM 版
 - connection_authentication 选项, 491
 - database_authentication 选项, 495
 - 关于, 70
 - 升级数据库, 74
 - 应用程序验证, 71
 - 开发应用程序, 70
 - 数据库验证, 71
 - 验证签名, 70
 - 验证连接, 72
- SQL Anywhere SNMP Extension Agent
 - 重新启动, 963
 - sasnmplib.ini 文件, 961
- 关于, 955
- 动态陷阱, 966
- 执行函数, 965
- 执行存储过程, 965
- 支持的 MIB, 957
- 支持的平台, 956
- 配置, 961
- SQL Anywhere VSS 写入程序
 - dbvss11.exe, 220, 829
- SQL Anywhere Web 版
 - 关于, 75
- SQL Anywhere 插件
 - 调试模式, 631
 - 使用, 630
 - 实体关系图, 632
 - 应用程序分析模式, 631
 - 概述选项卡, 634
 - 设计模式, 630
- SQL Anywhere 传送层安全
 - 关于, 1031
- SQL Anywhere 的本地化版本
 - 关于, 378
- SQL Anywhere 服务器示例
 - 使用, 313
- SQL Anywhere 归类算法 (SACA)
 - 关于, 392
- SQL Anywhere 环境变量
 - Unix, 344
 - 关于, 344
 - 在 Mac OS X 上设置, 344
 - 在 Unix 上发起, 344
 - 在 Windows 上设置, 344
 - 设置, 344
- SQL Anywhere 监控器 (见 监控器)
- SQL Anywhere 卷影复制服务 (VSS)
 - 备份类型, 829
- SQL Anywhere 控制台实用程序 [dbconsole]
 - Mac OS X 硬件要求, 688
 - 使用, 687
 - 启动, 687
 - 语法, 780
 - 软件更新, 690
 - 配置, 688
- SQL Anywhere 支持实用程序 [dbsupport]
 - SADIAGDIR 环境变量, 353
- SQL Anywhere 中的线程
 - 关于, 47

- SQLCONNECT 环境变量
 - 与 dbstop 实用程序一起使用, 784
 - 优先, 81
 - 说明, 362
 - 连接, 123
- SQLDA
 - ansi_blanks 选项, 480
- SQL Flagger
 - sql_flagger_error_level 选项, 545
 - sql_flagger_warning_level 选项, 546
- SQLPATH 环境变量
 - 说明, 363
- SQL Remote
 - SQL Remote 复制选项, 476
 - Windows Mobile 上不支持的功能, 340
 - 自动重命名事务日志, 848
 - 术语定义, 1112
- SQLREMOTE 环境变量
 - 说明, 364
- SQL 标准
 - Transact-SQL 兼容性选项, 473
 - UPDATE 语句, 753
- SQL 兼容性
 - 数据库选项, 472
- SQL 文件格式
 - Interactive SQL 输出, 680
- SQL 语句
 - Windows Mobile 上不支持的语句, 337
 - 在 Interactive SQL 中记录, 647
 - 在 Sybase Central 中记录, 42
 - 实用程序数据库, 28
 - 客户端/服务器, 387
 - 捕获最近准备, 227
 - 术语定义, 1112
- SQL 语句窗格
 - 说明, 639
- sr_date_format 选项
 - SQL Remote 选项, 547
- sr_time_format 选项
 - SQL Remote 选项, 548
- sr_timestamp_format 选项
 - SQL Remote 选项, 549
- SSL (见 传送层安全性)
- SSPI
 - 在 Windows 上进行 Kerberos 登录, 112
- StartDBPermission 属性
 - SQL Anywhere SNMP Extension Agent OID, 975
 - 服务器属性说明, 588
- START JAVA 语句
 - Windows Mobile 上不支持, 337
- StartLine 连接参数
 - 嵌入式数据库, 119
 - 常用选项, 43
 - 说明, 281
- StartTime 属性
 - SQL Anywhere SNMP Extension Agent OID, 975
 - 服务器属性说明, 588
- START 连接参数
 - 嵌入式数据库, 119
 - 常用选项, 43
 - 说明, 281
- StatementDescribes 属性
 - 数据库属性说明, 601
 - 连接属性说明, 566
- StatementPostAnnotatesSimple 属性
 - 数据库属性说明, 601
 - 连接属性说明, 566
- StatementPostAnnotatesSkipped 属性
 - 数据库属性说明, 601
 - 连接属性说明, 566
- StatementPostAnnotates 属性
 - 数据库属性说明, 601
 - 连接属性说明, 566
- STOP JAVA 语句
 - Windows Mobile 上不支持, 337
- StreamsUsed 属性
 - 服务器属性说明, 588
- string_rtruncation 选项
 - SQL Anywhere SNMP Extension Agent OID, 987
 - Transact-SQL 兼容性, 473
 - 说明, 549
 - 连接属性说明, 566
- subscribe_by_remote 选项
 - SQL Remote 复制选项, 476
 - 说明, 550
- subsume_row_locks 选项
 - SQL Anywhere SNMP Extension Agent OID, 987
 - 说明, 550
 - 连接属性说明, 566
- suppress_tds_debugging 选项
 - SQL Anywhere SNMP Extension Agent OID, 987
 - 说明, 551
 - 连接属性说明, 566
- SuppressWarnings 连接参数

-
- ODBC 连接参数说明, 714
 - Sybase Central
 - 校验表, 860
 - 校验数据库, 860
 - Mac OS X 硬件要求, 621
 - SQL Anywhere 插件, 630
 - 代码编辑器, 625
 - 停止数据库, 55
 - 关于, 620
 - 创建 Windows Mobile 数据库, 323
 - 创建数据库, 20
 - 创建服务, 60
 - 创建用户, 431
 - 创建组, 444
 - 删除数据库, 32
 - 向导在 Windows Mobile 上不受支持, 339
 - 启动, 621
 - 启动未连接的数据库, 54
 - 在 Windows Mobile 上运行数据库, 330
 - 在右窗格中自定义列, 624
 - 在组中添加用户, 444
 - 备份数据库, 827, 828
 - 复制数据库对象, 631
 - 审计, 1012
 - 文本完成, 683
 - 日志查看器, 630
 - 更改日志文件名, 16
 - 术语定义, 1113
 - 检索审计信息, 1013
 - 注册表设置, 374
 - 浏览, 622
 - 消除数据库, 32
 - 状态栏, 624
 - 管理 Windows 服务, 60
 - 记录 SQL 语句, 42
 - 设置数据库选项, 462
 - 软件更新, 690
 - 连接配置文件, 88
 - 通过经过验证的应用程序使用, 70
 - 配置快速启动程序, 686
 - 键盘快捷方式, 625
 - SYBASE-MIB.mib 文件
 - 关于, 956
 - SYBASE 环境变量
 - DSEdit, 1063
 - 说明, 365
 - sybinit 实用程序
 - 关于, 1060
 - sybping
 - 使用, 1075
 - synchronize_mirror_on_commit 选项
 - SQL Anywhere SNMP Extension Agent OID, 987
 - 说明, 551
 - 连接属性说明, 566
 - SyncTrunc 属性
 - SQL Anywhere SNMP Extension Agent OID, 984
 - 数据库属性说明, 601
 - SYS
 - 术语定义, 1113
 - SYSCOLAUTH
 - 统一的视图权限, 458
 - SYSCOLPERM
 - 系统视图权限, 458
 - SYSGROUP
 - 系统视图权限, 458
 - SYSGROUPS
 - 统一的视图权限, 458
 - Syslog
 - 用户 ID 针对, 205
 - SYSPROCAUTH
 - 统一的视图权限, 458
 - SYSPROCPERM
 - 系统视图权限, 458
 - SYSTABAUTH
 - 统一的视图权限, 458
 - SYSTABLEPERM
 - 系统视图权限, 458
 - SYSUSER
 - 统一的视图权限, 458
 - SYSUSERAUTH
 - 统一的视图权限, 458
 - SYSUSERAUTHORITY
 - 系统视图权限, 458
 - SYSUSERLIST
 - 统一的视图权限, 458
 - SYSUSERPERM
 - 兼容性视图权限, 458
 - SYSUSERPERMS
 - 兼容性视图权限, 458
 - SYS 组
 - 关于, 448
 - 散列
 - 术语定义, 1108
 - 删除
-

- ANSI 行为, 481
- dbspace, 27
- Kerberos 登录, 111
- Linux 服务, 769
- Transact-SQL 权限, 481
- 使用 Interactive SQL 删除行, 656
- 数据库, 32
- 数据库文件, 32
- 服务, 773
- 消除 [dberase] 实用程序, 722
- 用户, 440
- 组, 448
- 组中的用户, 445
- 表中的行, 656
- 集成登录, 101
- 删除的连接
 - SQL Anywhere SNMP Extension Agent 陷阱, 966
- 删除登录策略
 - 关于, 420
- 删除连接
 - 从数据库, 132
- 删除事务日志
 - 关于, 850
- 删除数据库
 - (参见 消除数据库)
 - 安全, 1018
- 商业证书颁发机构
 - 传送层安全, 1036
- 上部代码页
 - 关于, 385
- 上载
 - 术语定义, 1109
- 舍入
 - scale 选项, 543
- 设备跟踪
 - 术语定义, 1109
- 设计模式
 - 关于, 630
- 设置
 - 临时选项, 463
 - 数据库选项, 462
 - 轮询频率, 65
- 设置 [执行语句] 工具栏按钮
 - 关于, 642
- 设置传送层安全
 - 关于, 1035
- 设置数据库选项
 - 关于, 462
 - 设置自签名证书
 - 传送层安全, 1036
- 审计
 - conn_auditing 选项, 491
 - Sybase Central, 1012
 - Windows Mobile 上的数据库, 1029
 - 事务日志 [dblog] 实用程序操作, 1017
 - 关于, 1012
 - 启用, 1012
 - 安全功能, 1004
 - 恢复未提交的操作, 834
 - 控制, 484
 - 检索审计信息, 1013
 - 注释, 1015
 - 示例, 1015
 - 禁用, 1012
 - 翻译日志文件 [dbtran] 实用程序, 753
 - 翻译日志文件 [dbtran] 实用程序操作, 1017
 - 连接, 1013
- 生成
 - ECC 证书, 705
 - Interactive SQL 中的 sql 语句, 649
 - RSA 证书, 705
- 生成的连接条件
 - 术语定义, 1110
- 生成的数据库文档 (见 生成数据库文档)
- 生成实用程序 (见 后台启动服务器实用程序 [dbspawn])
- 生成数据库文档
 - 关于, 635
- 生成文档 (见 生成数据库文档)
- 升级
 - 数据库, 808
 - 经过验证的数据库, 74
- 升级实用程序 [dbupgrad]
 - 语法, 808
 - 退出代码, 809
- 升级数据库向导
 - Windows Mobile 上不支持, 339
- 实例化视图
 - materialized_view_optimization 选项, 515, 516
 - 术语定义, 1109
- 实体关系图
 - 在 SQL Anywhere 插件中查看, 632
- 实体关系选项卡
 - 使用, 632

实用程序

(参见 数据库实用程序)

重建 [rebuild], 760

重建 [rebuild] 语法, 760

dbisqlc 语法, 720

DSEdit, 1062

Interactive SQL [dbisql] 语法, 740

校验 [dbvalid] 语法, 810

Linux 服务 [dbsvc] 语法, 769

MobiLink 证书创建 [createcert] 语法, 705

MobiLink 证书查看器 [viewcert], 708

ping [dbping] 语法, 757

SQL Anywhere 控制台 [dbconsole] 实用程序, 780

SQL Anywhere 脚本执行 [dbrunsql], 761

Windows 服务 [dbsvc] 语法, 773

事务日志 [dblog] 审计, 1017

事务日志 [dblog] 语法, 792

使用配置文件, 695

信息 [dbinfo] 语法, 728

停止 [dbstop] 权限, 181

停止服务器 [dbstop] 语法, 783

初始化 [dbinit] 语法, 729

升级 [dbupgrad] 语法, 808

卸载 [dbunload] 语法, 795

后台启动服务器 [dbspawn] 语法, 781

在 Unix 上发起, 344

在配置文件中 使用条件分析, 696

备份 [dbbackup] 语法, 698

密钥对生成器 [createkey], 744

密钥对生成器 [createkey] 语法, 744

广播转发器 [dbns11] 语法, 703

支持 [dbsupport] 语法, 785

数据源 [dbdsn] 语法, 710

文件隐藏 [dbfhide] 语法, 724

日志传送管理器 [dbltm] 语法, 748

服务器授权 [dblic] 语法, 766

服务器枚举 [dblocate] 语法, 763

消除 [dberase] 语法, 722

版本诊断 [dbversion], 813

直方图 [dbhist] 语法, 726

简介, 695

翻译日志文件 [dbtran] 审计, 1017

翻译日志文件 [dbtran] 语法, 753

证书创建 [createcert] 语法, 705

证书查看器 [viewcert] 语法, 708

语言选择 [dblang] 语法, 745

通过经过验证的应用程序使用, 70

实用程序命令

权限, 188

实用程序数据库

util_db.ini 文件, 30

允许的 SQL 语句, 28

关于, 28

安全, 30

控制语句执行权限, 30

设置口令, 211

连接, 29

使用 dbinit 初始化数据库

关于, 729

使用信息

显示, 157

示例

从 Windows [开始] 菜单访问, 368

位置, 368

安装位置, 368

环境变量, 360

示例目录

关于, 368

示例数据库

Windows Mobile 上的两个版本, 313

启动 demo.db, 5

教程, 3

示例应用程序

ADO.NET 示例, 314

ESQL 示例, 316

ODBC 示例, 316

SQL Anywhere 服务器示例, 313

Windows Mobile, 313

世代号

术语定义, 1109

事件

创建调度向导, 867

调度, 864, 866

定义调度, 866

关于, 865

内部, 873

处理, 864

定义, 864

手工, 866

手工触发, 875, 876

数据库镜像, 882

条件, 866

生成数据库文档, 635

- 系统, 866
- 限制, 614
- 隐藏事件处理程序, 877
- 事件处理
 - 关于, 864
- 事件处理程序
 - 调试, 871
 - 事务行为, 874
 - 内部, 874
 - 定义, 864
 - 影响已授权连接, 874
 - 隐藏, 877
- 事件类型
 - 关于, 873
- 事件模型
 - 术语定义, 1109
- 事件日志
 - 取消消息, 42
- 事务
 - 事件处理程序行为, 874
 - 事务跨越多个日志文件时的恢复, 837
 - 关闭游标, 489
 - 分布式, 213, 214
 - 数据库镜像, 883
 - 术语定义, 1109
- 事务安全性
 - 数据库镜像, 883
 - 确保在数据库镜像中, 884
- 事务模式
 - 链接/非链接, 488
- 事务日志
 - a 数据库选项, 234
 - ad 数据库选项, 234
 - ar 数据库选项, 235
 - as 数据库选项, 236
 - f 恢复选项, 175
 - m 数据库选项, 238
 - m 服务器选项, 194
 - 备份的重命名, 848
 - 重命名备份副本, 849
 - delete_old_logs 选项, 502
 - 校验, 851
 - 使用重命名的事务日志备份, 849
 - >SQL Remote 自动重命名, 848
 - Windows Mobile, 321
 - 为复制服务器管理, 1084
 - 主键, 16
 - 事务日志 [dblog] 实用程序, 794
 - 介质故障, 842
 - 从介质故障中恢复, 841
 - 从多个中恢复, 837
 - 位置, 40
 - 使用原始事务日志备份, 825
 - 关于, 14
 - 分配空间, 26
 - 创建事务日志镜像, 22
 - 初始化 [dbinit] 实用程序, 729
 - 删除旧的, 792
 - 启动不使用, 175
 - 启动现有数据库的事务日志镜像, 16
 - 在备份过程中删除, 850
 - 在恢复时从数据库确定位置, 235
 - 在恢复时指定位置, 234
 - 在无 -m 选项的情况下运行, 729
 - 在无 -m 选项运行的情况下发出警告, 729
 - 在检查点之后删除, 194
 - 在检查点之后截断, 238
 - 备份 [dbbackup] 实用程序, 698
 - 备份的管理, 853
 - 大小, 16
 - 建议的位置, 14
 - 恢复期间应用, 234
 - 放置, 846
 - 数据库镜像, 901
 - 数据库镜像限制, 882
 - 日志传送管理器, 1072
 - 更改位置, 15
 - 未提交的更改, 834
 - 术语定义, 1109
 - 查找未完成的事务, 17
 - 活动备份, 821
 - 用 dberase 消除, 722
 - 管理, 502
 - 翻译实用程序, 755
 - 翻译日志文件 [dbtran] 实用程序, 753
 - 设置事务日志镜像文件名, 792
 - 选项, 46
 - 镜像和复制服务器, 1090
 - 限制大小, 869
- 事务日志 dbspace
 - 关于, 13
- 事务日志镜像
 - 与活动备份的区别, 822
 - 关于, 15

- 创建, 22
- 初始化 [dbinit] 实用程序, 729
- 启动, 16
- 建议的位置, 15
- 术语定义, 1110
- 目的, 15
- 事务日志镜像 dbspace
 - 关于, 13
- 事务日志实用程序 [dblog]
 - 审计, 1017
 - 语法, 792
 - 退出代码, 794
- 事务日志文件
 - 从 Sybase Central 更改名称, 16
- 事务完整性
 - 术语定义, 1110
- 视图
 - RESOURCE 特权, 427
 - 安全, 452
 - 安全功能, 1004
 - 所有者, 429
 - 授予权限, 435
 - 术语定义, 1109
 - 权限, 428
 - 生成数据库文档, 635
- 收藏夹列表
 - 关于, 644
- 手工事件
 - 创建, 875
- 首选服务器
 - 为数据库镜像指定, 897
- 首选项
 - SQL Anywhere 控制台 [dbconsole] 实用程序, 688
- 守护程序
 - ud 数据库服务器选项, 216
 - LTM, 748
 - 复制代理, 748
 - 将数据库服务器运行行为, 57
 - 日志传送管理器 [dbltn] 实用程序, 748
- 授权
 - 可执行文件, 767
 - 处理器许可, 766
 - 影响 -gm 服务器选项, 182
 - 影响 -gt 服务器选项, 185
 - 影响 -gtc 服务器选项, 186
 - 服务器授权 [dblic] 语法, 766
 - 每存取设备许可, 766
 - 连接限制和事件处理程序, 874
- 授权选项
 - 术语定义, 1110
- 授予
 - REMOTE 权限, 439
- 受保护的功能
 - secure_feature_key 选项, 544
 - 使用 -sf 指定, 207
 - 使用 -sk 指定安全功能密钥, 210
 - 关于, 1010
 - 术语定义, 1110
- 输入
 - Interactive SQL 中的多条语句, 643
 - Interactive SQL 命令, 641
- 属性
 - SQL Anywhere MIB 中的数据库属性, 983
 - SQL Anywhere MIB 中的服务器属性 OID, 975
 - 按字母顺序排序的数据库属性列表, 601
 - 按字母顺序排序的服务器属性列表, 588
 - 按字母顺序排序的连接属性列表, 566
 - 访问数据库属性, 601
 - 访问数据库服务器属性, 588
 - 访问连接属性, 566
- 术语表
 - SQL Anywhere 术语列表, 1095
- 数据操作语言
 - 术语定义, 1110
- 数据恢复
 - 关于, 817
- 数据库
 - 重建, 760
 - 重建后文件的大小减少, 804
 - 从 Sybase Central 进行校验, 860
 - 校验, 810, 820
 - 校验校验和, 810
 - Windows Mobile 上的安全, 1029
 - Windows Mobile 上的审计, 1029
 - Windows Mobile 上的用户标识, 1004
 - 在 Windows Mobile 上重建, 326
 - 为 Windows Mobile 创建, 322
 - 为 Windows Mobile 配置, 321
 - 为非英语语言创建, 401
 - 事务日志, 20
 - 从 Interactive SQL 连接, 86
 - 从 SQL 创建, 21
 - 从 SQL 初始化, 21

- 从 Sybase Central 创建, 20
- 从 Sybase Central 初始化, 20
- 从 Sybase Central 进行备份, 828
- 从 Sybase Central 连接, 86
- 从 Windows Mobile 设备消除, 328
- 从命令行创建, 22
- 从命令行消除, 32
- 从数据库断开连接, 132
- 使用 dbinit 创建, 729
- 使用 dbunload 卸载, 795
- 使用 dbupgrad 升级, 808
- 使用 Interactive SQL 在 Windows Mobile 上管理, 333
- 使用 Sybase Central 在 Windows Mobile 上管理, 330
- 信息, 728
- 停止, 54, 55, 783
- 停止权限, 181
- 健康和统计信息, 634
- 分配空间, 26
- 创建, 20
- 创建 dbspace, 24
- 初始化, 20
- 删除, 32
- 删除 dbspace, 27
- 加密, 1019, 1020
- 升级经过验证的, 74
- 卸载权限, 181
- 压缩加密的文件, 1019
- 变更 dbspace, 26
- 只读, 46, 204, 240
- 名称最大长度, 614
- 名称限制, 240
- 启动, 54
- 启动未连接的数据库, 54
- 启动权限, 179
- 备份, 817
- 复制到 Windows Mobile 设备, 325
- 复制服务器复制, 1091
- 多个文件, 23
- 大数据库, 23
- 字符集, 386
- 实用程序, 28, 694
- 恢复, 817
- 按字母顺序排序的属性列表, 601
- 故障排除连接, 124
- 文件兼容性, 20
- 更改归类, 402
- 最大大小, 614
- 最小大小, 729
- 术语定义, 1111
- 权限, 415
- 消除, 32
- 添加到现有服务, 64
- 用 dberase 消除, 722
- 禁用功能, 1010
- 自动停止, 177
- 装载权限, 181
- 解密, 1022
- 设置选项, 462
- 访问连接窗口, 87
- 连接, 79
- 连接到本地数据库, 117
- 连接方案, 116
- 通过 SQL Anywhere 控制台实用程序连接, 86
- 通过经过验证的应用程序使用, 70
- 页面使用情况, 728
- 验证, 495
- 数据库崩溃
 - 关于, 832
- 数据库大小
 - 限制, 614
- 数据库对象
 - 在 SQL Anywhere 插件中复制, 631
 - 术语定义, 1111
 - 确定 dbspace, 23
- 数据库对象权限
 - 关于, 428
- 数据库访问
 - 控制, 1006
 - 控制用户可以执行的任务, 1009
- 数据库服务器
 - SATMP 环境变量, 357
 - Windows Mobile, 329
 - 与 LDAP 一起使用, 138
 - 个人服务器与网络服务器之间的区别, 36
 - 临时文件位置, 12
 - 为镜像服务器指定替代名, 241
 - 作为守护程序运行, 57
 - 使用 dbping 进行查找, 130
 - 使用 FIPS 认可的高度加密算法, 176
 - 使用 UNC 连接参数停止, 282
 - 使用的内核数, 186
 - 停止, 53

- 关闭, 214
- 名称, 195
- 名称截断长度, 196
- 名称最大长度, 614
- 名称选项, 43
- 名称限制, 196
- 启动, 40
- 启动支持传送层安全, 1042
- 命令行, 148
- 在 Unix 上启动, 38
- 在 Vista 上运行, 38
- 在 Windows Mobile 上停止, 333
- 在 Windows Mobile 上启动, 318
- 在 Windows Vista 上自动启动, 264
- 在 Windows 操作系统上启动, 37
- 在后台运行, 57
- 安全, 1018
- 安静模式, 46
- 属性, 588
- 常用选项, 43
- 指定替代名, 242
- 按字母顺序排序的属性列表, 588
- 接口, 5
- 服务, 57
- 术语定义, 1111
- 查找, 126
- 监控, 913
- 禁止启动, 250
- 禁用数据库功能, 1010
- 窗口, 5
- 缺省, 195
- 自动启动, 126
- 记录操作, 40
- 运行, 3
- 进程并发水平, 50
- 连接到, 3
- 选项, 148
- 通过数据库镜像提供高可用性, 880
- 防止成为缺省, 223
- 静默, 46
- 高速缓存名称, 129
- 数据库服务器监控
 - 关于, 913
- 数据库服务器属性 (见 服务器属性)
 - 区分大小写, 588
- 数据库服务器外的审计操作
 - 关于, 1017
- 数据库服务器消息窗口
 - 保留最大化, 202
 - 关于, 40
 - 取消, 204
 - 取消性能消息, 203
 - 在 Linux 上使用, 219
 - 显示, 201
- 数据库服务器消息日志
 - 关于, 40
 - 截断, 199
 - 指定文件, 196
 - 获取名称, 196
 - 限制大小, 198
- 数据库管理实用程序
 - 关于, 694
- 数据库管理员
 - 术语定义, 1111
- 数据库归类
 - 关于, 394
- 数据库记录
 - 关于, 635
- 数据库加密
 - Windows Mobile, 322
 - 关于, 1019
- 数据库监控
 - 关于, 913
- 数据库镜像
 - sm 选项, 241
 - sn 选项, 242
 - xa 选项, 222
 - xf 选项, 223
 - xp 选项, 244
 - LOAD TABLE 语句限制, 882
 - SQL Anywhere SNMP Extension Agent 陷阱, 966
 - Windows Mobile 上不支持, 336
 - 不能截断事务日志, 882
 - 仅支持 TCP/IP 连接, 882
 - 从主服务器故障恢复, 900
 - 仲裁服务器角色, 883
 - 优点, 882
 - 停止数据库服务器, 900
 - 关于, 880
 - 关闭主服务器, 899
 - 同步模式, 883, 884
 - 同步状态, 885
 - 备份, 902
 - 客户端连接, 896

- 异步模式, 884
- 异步满页模式, 884
- 强制故障转移, 899
- 性能, 902
- 情形, 902
- 教程, 887, 891
- 日志文件, 901
- 法定数目, 881
- 状态信息文件, 885
- 确定主服务器, 897
- 系统事件, 901
- 角色切换, 880
- 设置 `synchronize_mirror_on_commit` 选项, 551
- 配置, 896
- 镜像服务器的只读访问, 898
- 限制, 882
- 需要网络数据库服务器, 882
- 首选服务器, 897
- 数据库镜像的优点
 - 关于, 882
- 数据库连接
 - `ping [dbping]` 实用程序, 757
 - 术语定义, 1111
- 数据库名
 - 区分大小写, 44
 - 最大长度, 614
 - 设置, 239
 - 选项, 43
- 数据库名称
 - 术语定义, 1111
- 数据库权限和特权
 - 关于, 422
- 数据库实用程序
 - (参见 实用程序)
 - 重建 [`rebuild`], 760
 - Interactive SQL [`dbisql`], 740
 - 校验 [`dbvalid`], 810
 - `ping [dbping]`, 757
 - SQL Anywhere 控制台 [`dbconsole`] 实用程序, 780
 - 事务日志 [`dblog`], 792
 - 信息 [`dbinfo`], 728
 - 停止服务器 [`dbstop`], 783
 - 初始化 [`dbinit`], 729
 - 升级 [`dbupgrad`], 808
 - 卸载 [`dbunload`], 795
 - 后台启动服务器 [`dbspawn`], 781
 - 备份 [`dbbackup`], 698
 - 数据库连接, 123
 - 数据源 [`dbdsn`], 710
 - 文件隐藏 [`dbfhide`], 724
 - 日志传送管理器 [`dbltm`] 实用程序, 748
 - 服务 [`dbsvc`], 773
 - 服务器授权 [`dblic`], 766
 - 服务器枚举 [`dblocate`], 763
 - 版本诊断 [`dbversion`], 813
 - 直方图 [`dbhist`], 726
 - 翻译日志文件 [`dbtran`], 753
 - 脚本执行 [`dbrunsql`], 761
 - 语言选择 [`dblang`], 745
- 数据库属性
 - SQL Anywhere SNMP Extension Agent 的 OID, 983
 - 使用 SQL Anywhere SNMP Extension Agent 检索, 964
 - 使用 SQL Anywhere SNMP Extension Agent 设置, 965
 - 区分大小写, 601
 - 报告, 757
 - 按字母顺序排序的列表, 601
- 数据库损坏
 - 关于, 858
- 数据库所有者
 - 术语定义, 1112
- 数据库统计信息
 - SQL Anywhere MIB, 980
 - SQL Anywhere SNMP Extension Agent 的 OID, 980
 - 使用 SQL Anywhere SNMP Extension Agent 检索, 964
- 数据库文档向导
 - 关于, 635
- 数据库文件
 - NDS, 155
 - UNC 文件名, 155
 - 介质故障, 833
 - 位置, 40
 - 使用 `dbinit` 加密, 729
 - 关于, 12
 - 加密, 1019
 - 备份, 852
 - 安全, 1005
 - 最大大小, 614
 - 最小大小, 729

- 术语定义, 1112
- 用 `dberase` 消除, 722
- 路径, 155
- 限制, 24
- 数据库文件加密
 - 初始化 [dbinit] 实用程序, 729
- 数据库信息
 - 使用 `dbinfo` 获取, 728
- 数据库选项
 - ASE 兼容性选项, 473
 - Interactive SQL 选项, 667
 - `isolation_level` 数据库选项, 508
 - Open Client, 1067
 - `read_past_deleted`, 538
 - SQL Anywhere MobiLink 客户端设置, 475
 - SQL Anywhere SNMP Extension Agent 的 OID, 987
 - SQL Remote 复制选项, 476
 - Transact-SQL 兼容性选项, 473
 - `truncate_timestamp_values`, 556
 - 作用域和持续时间, 463
 - 使用 SQL Anywhere SNMP Extension Agent 检索, 964
 - 使用 SQL Anywhere SNMP Extension Agent 设置, 965
 - 关于, 462
 - 分类, 466
 - 初始设置, 465
 - 删除设置, 466
 - 区分大小写, 463
 - 启动设置, 1067
 - 在 Interactive SQL 中设置, 667
 - 复制代理, 477
 - 按字母顺序排序的 SQL Remote 选项列表, 476
 - 按字母顺序排序的 Transact-SQL 兼容性选项列表, 473
 - 按字母顺序排序的同步选项列表, 475
 - 按字母顺序排序的复制代理选项列表, 477
 - 按字母顺序排序的数据库兼容性选项列表, 472
 - 按字母顺序排序的数据库选项列表, 466
 - 查找值, 464
 - 概述, 461
 - 监控设置, 465
 - 设置, 462
- 数据库页
 - 为进行高速缓存预热进行收集, 160
 - 预热数据库高速缓存, 165
- 数据库页面
 - 显示大小, 728
- 数据库中的 Java
 - cp 服务器选项, 164
- 数据类型
 - 支持的, 1084
 - 术语定义, 1110
 - 限制, 614
- 数据类型转换
 - 错误, 493
- 数据立方体
 - 术语定义, 1111
- 数据源
 - Interactive SQL [dbisql] 实用程序, 740
 - Mac OS X, 93
 - ODBC, 90
 - ODBC 连接参数, 714
 - SQL Anywhere 控制台 [dbconsole] 实用程序, 780
 - Unix, 95
 - 为 Windows Mobile 创建, 319
 - 使用 [连接] 窗口生成, 91
 - 使用 `dbdsn` 创建 ODBC, 710
 - 使用 ODBC 管理器生成, 91
 - 关于, 90
 - 在 Windows Mobile 上使用, 94
 - 嵌入式 SQL, 90
 - 数据源 [dbdsn] 实用程序, 710
 - 文件, 94
 - 示例, 119
 - 连接参数, 80
- 数据源实用程序 [dbdsn]
 - 在 Vista 上可能需要权限提升, 38
 - 系统信息文件, 713
 - 语法, 710
 - 退出代码, 713
- 数字
 - 最大位数, 533
- 数字精度
 - 数据库选项, 533
- 数字签名
 - MobiLink 传送层安全, 1049
 - SQL Anywhere 传送层安全, 1043
- 数字证书
 - 传送层安全, 1036
- 死锁
 - Deadlock 系统事件, 868

- log_deadlocks 选项, 511
- 术语定义, 1112
- 死锁报告
 - log_deadlocks 选项, 511
- 四进制标点符号区分
 - 不区分大小写和重音的数据库, 739
- 搜索
 - Sybase Central, 624
 - 数据库, 624
- 搜索条件
 - user_estimates 选项, 559
- 碎片整理
 - 性能, 26
- 缩进
 - Interactive SQL 快捷方式, 663
- 索引
 - 术语定义, 1113
 - 限制, 614
- 锁定
 - 术语定义, 1112
 - 镜像数据库, 898
- 锁定冲突
 - blocking 选项, 486
 - blocking_timeout 选项, 487
- 所有者
 - 关于, 429

T

TCP/IP

- x 服务器选项, 221
- BroadcastListener [BLISTENER] 协议选项, 286
- ClientPort [CPORT] 协议选项, 290
- HOST [IP] 协议选项, 292
- IPv6 支持, 135
- LDAP 协议选项, 295
- Open Server, 1060
- ServerPort [PORT] 协议选项, 303
- Windows, 136
 - 关于, 135
 - 协议选项, 284
 - 受支持的协议, 134
 - 启动, 51
 - 地址, 1064
 - 对客户端/服务器通信进行加密, 137
 - 性能, 136
 - 故障排除, 143
 - 数据库镜像时所必需, 882

- 服务器配置, 284
- 用于 SQL Anywhere 客户端的 MobiLink TLS, 1051
- 用于 UltraLite 客户端的 MobiLink TLS , 1052
- 穿过防火墙定位服务器, 765
- 穿过防火墙进行连接, 136
- 端口号, 1061
- 端口标识, 303
- 防火墙和 LDAP 服务器, 138
- TCP/IP 传送层安全
 - MobiLink , 1051
- TcpIpAddresses 属性
 - SQL Anywhere SNMP Extension Agent OID, 975
 - 服务器属性说明, 588
- tds_empty_string_is_null 选项
 - SQL Anywhere SNMP Extension Agent OID, 987
 - 说明, 552
 - 连接属性说明, 566
- TDS 通信协议
 - 关于, 1058
- TDS 协议选项
 - 说明, 305
- Telnet
 - 测试网络, 68
- temp_space_limit_check 选项
 - SQL Anywhere SNMP Extension Agent OID, 987
 - 说明, 552
 - 连接属性说明, 566
- TempDir 属性
 - SQL Anywhere SNMP Extension Agent OID, 975
 - 服务器属性说明, 588
- TempDiskSpace 系统事件
 - 说明, 868
- TempFileName 属性
 - SQL Anywhere SNMP Extension Agent OID, 984
 - 数据库属性说明, 601
- TempTablePages 属性
 - SQL Anywhere SNMP Extension Agent OID, 980
 - 数据库属性说明, 601
 - 连接属性说明, 566
- TEMP 环境变量
 - Windows Mobile, 374
 - 磁盘空间, 68
 - 说明, 366
- 调试
 - debug_messages 选项, 498
 - Interactive SQL 中的 SQL 语句, 641

-
- SQL 脚本, 740
 - Web 服务客户端, 229
 - 事件处理程序, 871
 - 关于, 631
 - time_format 选项
 - ASE 兼容性, 473
 - Open Client, 1067
 - SQL Anywhere SNMP Extension Agent OID, 987
 - Transact-SQL 兼容性, 473
 - 说明, 553
 - 连接属性说明, 566
 - time_zone_adjustment 选项
 - SQL Anywhere SNMP Extension Agent OID, 987
 - 说明, 554
 - 连接属性说明, 566
 - Timeout 协议选项
 - 说明, 306
 - timestamp_format 选项
 - ASE 兼容性, 473
 - Open Client, 1067
 - SQL Anywhere SNMP Extension Agent OID, 987
 - Transact-SQL 兼容性, 473
 - 说明, 554
 - 连接属性说明, 566
 - TIMESTAMP 数据类型
 - default_timestamp_increment 选项, 500
 - 比较, 500
 - TimeZoneAdjustment 属性
 - SQL Anywhere SNMP Extension Agent OID, 975
 - 服务器属性说明, 588
 - 连接属性说明, 566
 - TLS
 - MobiLink 客户端 (SQL Anywhere), 1050
 - MobiLink 客户端 (UltraLite), 1052
 - Windows Mobile 上不支持, 338
 - 使用端对端加密的 MobiLink, 1047
 - 关于, 1031
 - 支持, 1033
 - tls_type 协议选项
 - dbeng11 -ec, 171
 - dbsrv11 -ec, 171
 - TLS 同步
 - 关于, 1031
 - TLS 支持
 - 关于, 1033
 - TMPDIR 环境变量
 - Windows Mobile, 374
 - 说明, 366
 - TMP 环境变量
 - Windows Mobile, 374
 - 说明, 366
 - TotalBuffers 属性
 - SQL Anywhere SNMP Extension Agent OID, 972
 - 服务器属性说明, 588
 - TO 协议选项
 - 说明, 306
 - TransactionStartTime 属性
 - 连接属性说明, 566
 - Transact-SQL
 - allow_nulls_by_default 选项, 477
 - NULL 行为, 484
 - quoted_identifier 选项, 537
 - 兼容性数据库选项, 472
 - 兼容性选项, 473
 - 列 NULL 兼容性, 537
 - 删除权限, 481
 - 更新权限, 481
 - 查询编辑器不支持, 651
 - TranslationDLL 连接参数
 - ODBC 连接参数说明, 714
 - TranslationName 连接参数
 - ODBC 连接参数说明, 714
 - TranslationOption 连接参数
 - ODBC 连接参数说明, 714
 - TriggerPages 属性
 - SQL Anywhere SNMP Extension Agent OID, 980
 - 数据库属性说明, 601
 - truncate_timestamp_values 选项
 - SQL Anywhere SNMP Extension Agent OID, 987
 - 用于 MobiLink 同步, 556
 - 说明, 556
 - 连接属性说明, 566
 - truncation_length 选项
 - Interactive SQL 设置, 667
 - 说明, 682
 - trusted_certificates 协议选项
 - MobiLink 传送层安全, 1050
 - 说明, 307
 - tsql_outer_joins 选项
 - ASE 兼容性, 473
 - SQL Anywhere SNMP Extension Agent OID, 987
 - Transact-SQL 兼容性, 473
 - 说明, 557
 - 连接属性说明, 566
-

- tsql_variables 选项
 - ASE 兼容性, 473
 - Open Client, 1067
 - SQL Anywhere SNMP Extension Agent OID, 987
 - Transact-SQL 兼容性, 473
 - 说明, 557
 - 连接属性说明, 566
- 特权
 - BACKUP, 424
 - DBA, 425
 - PROFILE, 425
 - READCLIENTFILE, 426
 - READFILE, 426
 - REMOTE DBA, 427
 - RESOURCE, 427
 - VALIDATE, 427
 - WRITECLIENTFILE, 427
 - 关于, 424
 - 撤消, 439
 - 管理, 415
 - 继承, 422
- 提供程序
 - MSDASQL, 97
 - OLE DB, 97
 - SAOLEDB, 97
- 提交
 - cooperative_commit_timeout 选项, 493
 - cooperative_commits 选项, 494
 - delayed_commit_timeout 选项, 501
 - delayed_commits 选项, 501
 - 错误报告给 iAnywhere, 76
- 提示
 - 连接参数, 81
- 替代服务器名
 - sn 选项, 242
- 替代字符
 - on_charset_conversion_failure 选项, 527
- 添加
 - ECC 和 RSA 证书, 705
 - Interactive SQL 中的新行, 655
 - 监控器用户, 942
- 条件分析
 - 配置文件, 696
- 停止
 - Windows Mobile 上的服务器, 333
 - 数据库, 783
 - 数据库 (ASTOP), 251
 - 停止服务器实用程序 [dbstop]
 - 与 SQLCONNECT 一起使用, 784
 - 使用, 53
 - 权限, 181
 - 语法, 783
 - 退出代码, 784
 - 停止数据库
 - 关于, 55
 - 停止数据库服务器
 - 镜像系统, 900
 - 通告程序
 - 术语定义, 1113
 - 通信
 - ec 服务器选项, 171
 - DatabaseKey [DBKEY] 连接参数, 259
 - Encryption [ENC] 连接参数, 265
 - 调试, 226
 - 关于, 133
 - 协议选项, 284
 - 受支持的, 134
 - 故障排除, 143
 - 数据库服务器, 221
 - 通信参数 (勿连接参数)
 - 通信流
 - 术语定义, 1113
 - 通信压缩
 - Compress [COMP] 连接参数, 255
 - CompressionThreshold [COMPTH] 连接参数, 256
 - 同步
 - 重建加密的数据库, 807
 - delete_old_logs 选项, 502
 - 传送层安全, 1031
 - 备份, 847
 - 数据库选项, 475
 - 数据库镜像, 883
 - 术语定义, 1113
 - 设置 default_timestamp_increment, 500
 - 设置 truncate_timestamp_values, 556
 - 同步模式
 - 数据库镜像, 883, 884
 - 同步状态
 - 数据库镜像, 885
 - 统计信息
 - [概述] 选项卡, 634
 - SQL Anywhere MIB 中的数据库统计信息, 980
 - SQL Anywhere SNMP Extension Agent 的服务器统计信息 OID, 972

- 统一数据库
 - 术语定义, 1113
- 图标
 - 此帮助文档中使用的, xv
 - 用于正在运行的服务, 63
- 图形式计划
 - 保存, 652
 - 查看, 652
- 土耳其语数据库
 - 不区分大小写的数据库, 412
 - 创建, 401
 - 区分大小写, 411
- 推式请求
 - 术语定义, 1114
- 推式通知
 - 术语定义, 1114
- 退出代码
 - 重建 [rebuild] 实用程序, 760
 - Interactive SQL [dbisql] 实用程序, 742
 - 校验实用程序 (dbvalid), 812
 - ping [dbping] 实用程序, 759
 - Windows 服务 [dbsvc] 实用程序, 777
 - 事务日志 [dblog] 实用程序, 794
 - 信息 [dbinfo] 实用程序, 728
 - 停止服务器 [dbstop] 实用程序, 784
 - 初始化 [dbinit] 实用程序, 739
 - 升级 [dbupgrad] 实用程序, 809
 - 卸载 [dbunload] 实用程序, 805
 - 后台启动服务器 [dbspawn] 实用程序, 781
 - 备份 [dbbackup] 实用程序, 702
 - 数据源 [dbdsn] 实用程序, 713
 - 日志传送管理器 [dbltn] 实用程序, 749
 - 服务器授权 [dblic] 实用程序, 767
 - 服务器枚举 [dblocate] 实用程序, 765
 - 消除 [dberase] 实用程序, 723
 - 直方图 [dbhist] 实用程序, 727
 - 翻译日志文件 [dbtran] 实用程序, 756
 - 语言 [dblang] 实用程序, 746
- 脱机备份
 - 关于, 817, 819
- 脱机事务日志
 - 备份, 854
- 椭圆曲线证书
 - 创建, 705
 - 查看, 708

U

- UAC
 - 在 Vista 上运行 SQL Anywhere, 38
- UCA
 - 关于, 393
 - 用于单字节字符集, 393
- UCA 归类
 - 瑞典学院标准, 398
 - 瑞典语区域设置, 398
- UID 连接参数
 - 说明, 283
- ulcond11 实用程序
 - 在 Vista 上可能需要权限提升, 38
- UltraLite
 - MobiLink 传送层安全, 1052
 - 术语定义, 1114
- UltraLite 客户端
 - TLS, 1052
- UltraLite 运行时
 - 术语定义, 1114
- UncommitOp 属性
 - 连接属性说明, 566
- Unconditional 连接参数
 - 说明, 282
- UNC 连接参数
 - 说明, 282
- Unicode 归类算法 (UCA)
 - 关于, 393
- Unicode 字符集
 - 关于, 392
- UniqueClientAddresses 属性
 - SQL Anywhere SNMP Extension Agent OID, 972
 - 服务器属性说明, 588
- Unix
 - IPv6 支持, 135
 - LD_LIBRARY_PATH 环境变量, 347
 - ODBC 支持, 95
 - ODBC_INI 环境变量, 350
 - ODBCHOME 环境变量, 349
 - ODBCINI 环境变量, 350
 - SATMP 环境变量, 366
 - 临时文件, 357
 - 临时文件权限, 357
 - 保护共享内存连接安全, 1005
 - 发起文件, 344
 - 启动 dbconsole, 688

- 启动 Interactive SQL, 637, 638
 - 启动数据库服务器, 38
 - 定位文件, 372
 - 将 Ping 实用程序用于线程连接库, 757
 - 授权可执行文件, 767
 - 系统信息文件, 95
 - 线程行为, 47
 - 缺省字符集, 391
 - 设置环境变量, 344
 - 选择归类, 409
 - 高速缓存大小, 158
 - Unix 上的任务
 - 关于, 47
 - UNLOAD TABLE 语句
 - WRITECLIENTFILE 特权, 427
 - 安全, 1018
 - UNLOAD 语句
 - 安全, 1018
 - UnschReq 属性
 - SQL Anywhere SNMP Extension Agent OID, 972
 - 服务器属性说明, 588
 - updatable_statement_isolation 选项
 - SQL Anywhere SNMP Extension Agent OID, 987
 - 说明, 557
 - 连接属性说明, 566
 - update_statistics 选项
 - SQL Anywhere SNMP Extension Agent OID, 987
 - 说明, 558
 - 连接属性说明, 566
 - update_timeout 参数
 - LDAP, 139
 - UPDATE 权限
 - 关于, 428
 - 授予, 434
 - UPDATE 语句
 - LTM 支持的操作, 1084
 - 在 Interactive SQL 中生成, 649
 - 字符串的截断, 549
 - upgrade_database_capability 选项
 - SQL Anywhere SNMP Extension Agent OID, 987
 - 连接属性说明, 566
 - user_estimates 选项
 - SQL Anywhere SNMP Extension Agent OID, 987
 - 说明, 559
 - 连接属性说明, 566
 - UserAppInfo 属性
 - 连接属性说明, 566
 - Userid 连接参数
 - 说明, 283
 - UserID 属性
 - 连接属性说明, 566
 - UTF-8
 - 用于数据库, 729
 - util_db.ini
 - 关于, 30
 - UtilCmdsPermitted 属性
 - 连接属性说明, 566
 - utility_db
 - 为实用程序数据库保留的名称, 28
 - 控制语句执行权限, 30
 - 连接, 29
- ## V
- VALIDATE 特权
 - 不可继承, 427
 - 关于, 427
 - 授予, 433
 - VCS 代理
 - SADatabase, 907
 - SAServer, 906
 - verify_all_columns 选项
 - SQL Remote 复制选项, 476
 - 说明, 560
 - verify_password_function 选项
 - SQL Anywhere SNMP Extension Agent OID, 987
 - 说明, 560
 - 连接属性说明, 566
 - 验证口令, 1007
 - verify_threshold 选项
 - SQL Remote 复制选项, 476
 - 说明, 563
 - VerifyServerName 协议选项
 - 说明, 308
 - VERIFY 协议选项
 - 说明, 308
 - Veritas 聚簇服务器
 - 与 SQL Anywhere 配合使用, 905
 - Veritas 聚簇服务器代理
 - 关于, 905
 - VersionStorePages 属性
 - SQL Anywhere SNMP Extension Agent OID, 980
 - 数据库属性说明, 601
 - viewcert 实用程序
 - 用法, 1036

-
- 语法, 708
 - ViewPages 属性
 - SQL Anywhere SNMP Extension Agent OID, 980
 - 数据库属性说明, 601
 - Vista
 - ActiveSync, 39
 - dbelevat11.exe, 39
 - DisableMultiRowFetch 连接参数, 264
 - SQL Anywhere 已提升的操作代理, 38
 - Windows 移动设备中心, 39
 - 使用 AWE 高速缓存, 39
 - 使用 dbconsole 实用程序来监控数据库服务器, 39
 - 在 Vista 上运行 SQL Anywhere, 38
 - 已提升的操作代理, 38
 - 服务没有与桌面交互, 39
 - 用户帐户控制, 38
 - 签名的可执行文件, 39
 - 部署注意事项, 39
 - VSS
 - SQL Anywhere 卷影复制服务, 829
 - 数据库服务器, 220
 - VSS 写入程序
 - dbvss11.exe, 829
 - W**
 - wait_for_commit 选项
 - SQL Anywhere SNMP Extension Agent OID, 987
 - 说明, 564
 - 连接属性说明, 566
 - WaitStartTime
 - 连接属性说明, 566
 - WaitType
 - 连接属性说明, 566
 - WebClientLogFile 属性
 - SQL Anywhere SNMP Extension Agent OID, 975
 - 服务器属性说明, 588
 - WebClientLogging 属性
 - SQL Anywhere SNMP Extension Agent OID, 975
 - 服务器属性说明, 588
 - webservice_namespace_host 选项
 - SQL Anywhere SNMP Extension Agent OID, 987
 - 说明, 564
 - 连接属性说明, 566
 - Web 版
 - 关于, 75
 - Web 服务
 - webservice_namespace_host 选项, 564
 - 启动支持传送层安全, 1046
 - 数据库服务器配置, 224
 - web 服务客户端
 - certificate_company 协议选项, 287
 - certificate_name 协议选项, 288
 - certificate_unit 协议选项, 289
 - trusted_certificates 协议选项, 307
 - Web 服务客户端
 - 记录, 229
 - Web 服务客户端日志文件
 - 设置名称, 229
 - Web 服务器
 - 启动支持传送层安全, 1046
 - 在镜像系统中不受支持, 882
 - Web 开发
 - Web 版, 75
 - 未调度的请求 (*见* ReqStatus 属性) (*见* UnschReq 属性)
 - Windows
 - (*参见* Windows 2000)
 - (*参见* Windows 2003)
 - (*参见* Windows Mobile)
 - (*参见* Windows XP)
 - IPv6 支持, 135
 - TCP/IP, 136
 - 事件日志, 42
 - 代码页, 386
 - 启动数据库服务器, 37
 - 字符集, 386
 - 安装 SNMP, 961
 - 安装注册表设置, 374
 - 定位文件, 370
 - 服务, 59
 - 术语定义, 1116
 - 线程行为, 48
 - 缺省字符集, 391
 - 选择归类, 408
 - 限制 AWE 高速缓存大小, 167
 - 集成登录, 99
 - 高速缓存大小, 158
 - Windows 2000
 - IPv6 支持, 135
 - 安装 SNMP, 961
 - 集成登录, 99
 - Windows 2003
 - 安装 SNMP, 961
-

- Windows CE (见 Windows Mobile)
- Windows MIT Kerberos 客户端
 - keytab 文件, 107
- Windows Mobile
 - ALTER DATABASE 语句限制, 337
 - BACKUP 语句限制, 337
 - 重建数据库, 326
 - CREATE DATABASE 语句, 337
 - CREATE EVENT 语句限制, 337
 - CREATE FUNCTION 语句限制, 337
 - CREATE TABLE 语句限制, 337
 - ESQL 示例, 316
 - ICU, 310
 - jConnect, 322
 - ODBC 示例, 316
 - 缺省情况下启用校验和, 859
 - SQL Anywhere 安装要求, 310
 - Windows Mobile 5.0 for Smartphone 的限制, 311
 - Windows Mobile 上有限支持创建维护计划向导, 339
 - 不支持 -? 服务器选项, 338
 - 不支持 -cm 服务器选项, 338
 - 不支持 -cw 选项, 338
 - 不支持 -gb 选项, 338
 - 不支持 -ge 选项, 338
 - 不支持 -qi 选项, 338
 - 不支持 -s 选项, 338
 - 不支持 -tmf 选项, 338
 - 不支持 -tmt 选项, 338
 - 不支持 -u 选项, 338
 - 不支持 -ua 选项, 338
 - 不支持 -uc 选项, 338
 - 不支持 -ud 选项, 338
 - 不支持 -uf 选项, 338
 - 不支持 -ui 选项, 338
 - 不支持 -ut 选项, 338
 - 不支持 -ux 选项, 338
 - 不支持 -xp 选项, 338
 - 不支持 @data 选项, 338
 - 不支持 CREATE EXISTING TABLE 语句, 337
 - 不支持 CREATE EXTERNLOGIN 语句, 337
 - 不支持 CREATE SERVER 语句, 337
 - 不支持 dbextract, 340
 - 不支持 DROP DATABASE 语句, 337
 - 不支持 DROP SERVER 语句, 337
 - 不支持 iAnywhere JDBC 驱动程序, 336
 - 不支持 INSTALL JAVA 语句, 337
 - 不支持 Kerberos, 336
 - 不支持 LDAP 验证, 336
 - 不支持 Open Client, 336
 - 不支持 REMOVE JAVA 语句, 337
 - 不支持 REORGANIZE TABLE 语句, 337
 - 不支持 START JAVA 语句, 337
 - 不支持 STOP JAVA 语句, 337
 - 不支持个人服务器, 336
 - 不支持创建数据库向导, 339
 - 不支持创建服务向导, 339
 - 不支持升级数据库向导, 339
 - 不支持卸载数据库向导, 339
 - 不支持备份数据库向导, 339
 - 不支持外部存储过程, 336
 - 不支持并行备份, 336
 - 不支持恢复数据库向导, 339
 - 不支持抽取实用程序, 340
 - 不支持数据库镜像, 336
 - 不支持更改日志文件设置向导, 339
 - 不支持消除数据库向导, 339
 - 不支持的 SQL Anywhere 功能, 336
 - 不支持的 SQL Remote 功能, 340
 - 不支持的 SQL 语句, 337
 - 不支持的数据库服务器选项, 338
 - 不支持的管理工具, 336
 - 不支持目录访问服务器, 336
 - 不支持翻译日志文件向导, 339
 - 不支持迁移数据库向导, 339
 - 不支持远程数据访问, 336
 - 事务日志, 321
 - 从 Interactive SQL 管理数据库, 333
 - 从 Sybase Central 运行数据库, 330
 - 从桌面连接, 85
 - 从计算机连接, 318
 - 使用 .NET Compact Framework, 311
 - 使用 ADO.NET 示例, 314
 - 使用 ODBC 数据源, 94
 - 使用 SQL Anywhere, 309
 - 使用管理实用程序, 330
 - 停止服务器, 333
 - 创建 ODBC 数据源, 319
 - 创建数据库, 322
 - 加密, 322
 - 受限制的 jConnect 功能, 336
 - 受限制的 ODBC 功能, 336
 - 向导在 Windows Mobile 上不受支持, 339
 - 启动多个数据库, 331

- 启动服务器, 331
- 子目录, 368
- 存储卡, 310
- 安全, 1029
- 安装, 368
- 定位文件, 371
- 审计, 1029
- 将数据库复制到设备, 325
- 应用程序分析限制, 336
- 数据库加密, 1029
- 数据库服务器, 329
- 数据库服务器选项, 1029
- 文件位置, 368
- 有限归类定制支持, 321
- 服务器启动选项窗口, 329
- 术语定义, 1116
- 消除数据库, 328
- 用户授权, 1004
- 用户标识, 1004
- 示例应用程序, 313
- 示例数据库, 313
- 设备安全, 1029
- 设置 TEMP 文件, 374
- 通信加密, 1030
- 配置 SQL Anywhere, 310
- 配置数据库, 321
- Windows Mobile 5.0
 - smartphone 限制, 311
- Windows Vista
 - DisableMultiRowFetch 连接参数, 264
 - 安装 SNMP, 961
- Windows XP
 - 安装 SNMP, 961
 - 集成登录, 99
- Windows 服务
 - Windows 服务管理器, 65
 - 了解, 59
 - 依赖性, 66, 67
 - 停止, 65
 - 创建, 60
 - 删除, 61
 - 参数, 61
 - 可执行文件, 64
 - 启动, 65
 - 启动选项, 62
 - 启动顺序, 67
 - 多个, 66
 - 帐户选项, 63
 - 数据库服务器, 57
 - 桌面上的图标, 63
 - 注册表设置, 373
 - 添加新数据库, 64
 - 符合条件的程序, 59
 - 管理, 60
 - 组, 66
 - 轮询, 65
 - 选项, 62
 - 配置, 61
- Windows 服务管理器
 - 关于, 65
- Windows 和 Linux 上的任务
 - 关于, 48
- Windows 性能监控器
 - 控制监控的数据库数量, 193
 - 控制监控的连接数量, 193
 - 禁止创建共享内存, 193
- Windows 用户组
 - 集成登录, 102
- Winsock
 - 在 Windows 上使用 TCP/IP, 136
- WITH GRANT OPTION 子句
 - 使用, 437
- WRITECLIENTFILE 特权
 - 关于, 427
 - 可继承, 427
 - 授予, 433
- 外表
 - 术语定义, 1114
- 外部存储过程
 - Windows Mobile 上不支持, 336
- 外部登录
 - 术语定义, 1114
- 外部函数
 - 堆栈大小, 180
- 外部卸载
 - 使用, 805
- 外键
 - 术语定义, 1114
- 外键约束
 - 术语定义, 1115
- 外连接
 - 术语定义, 1115
- 完全备份
 - 执行, 820

- 术语定义, 1115
- 完整性
 - 术语定义, 1115
- 网关
 - 术语定义, 1115
- 网络参数
 - (参见 连接参数)
- 网络服务器
 - 传送层安全, 1042
 - 关于, 36
 - 术语定义, 1115
 - 软件要求, 36
 - 连接, 120
- 网络服务器监控器
 - (参见 SQL Anywhere 控制台实用程序)
 - 使用, 687
 - 语法, 780
- 网络附加存储
 - 存储数据库文件, 14
- 网络连接
 - 选项, 51
- 网络驱动器
 - 数据库文件, 40
- 网络适配器
 - 驱动程序, 143
- 网络数据库服务器 (见 网络服务器)
- 网络通信
 - sasrv.ini 文件, 68
 - 调试启动问题, 68
 - 命令行选项, 284
 - 故障排除, 68, 143
- 网络协议
 - dbeng11 -x 选项, 221
 - dbsrv11 -x 选项, 221
 - 关于, 133
 - 受支持的, 134
 - 客户端选项, 284
 - 按字母顺序排序的列表, 284
 - 故障排除, 143
 - 术语定义, 1116
 - 网络连接, 120
- 网络协议选项
 - 数据库服务器, 247
- 唯一约束
 - 术语定义, 1116
- 维护版本
 - 术语定义, 1116
- 维护计划
 - 关于, 845
 - 报告, 845
- 维护计划报告
 - 关于, 845
- 维护用户
 - 主站点, 1075
 - 复制站点, 1077
 - 用户 ID, 1081
- 尾随空格
 - 在连接字符串内使用, 81
- 未提交的错误报告
 - 查看, 76
 - 监控器, 930
 - 监控器警告, 949
- 位数组
 - 术语定义, 1116
- 谓词
 - 术语定义, 1116
- 文本计划
 - 使用计划查看器的 UltraLite, 651
- 文本完成
 - 使用, 683
 - 配置, 684
 - 键盘快捷方式, 684
- 文本文件格式
 - Interactive SQL 输入, 672
 - Interactive SQL 输出, 680
- 文档
 - SQL Anywhere, xii
 - 约定, xiii
 - 记录数据库, 635
- 文件
 - 从 Interactive SQL 使用源代码控制, 658
 - 从 Interactive SQL 更新, 662
 - 从 Interactive SQL 签入, 662
 - 从 Interactive SQL 签出, 661
 - 位置, 370
 - 配置 Interactive SQL 源代码控制, 658
- 文件大小
 - 重建后减少, 804
- 文件定义数据库
 - 术语定义, 1116
- 文件夹列表
 - 显示 Sybase Central, 622
- 文件数据源
 - 创建, 94

- 文件位置
 - Windows Mobile, 368
- 文件隐藏实用程序 [dbfhide]
 - 语法, 724
- 问题
 - 字符集, 382
- 无法初始化任何通信链接错误
 - 诊断原因, 145
- 无法启动 -- 找不到服务器
 - 诊断原因, 144
- 物理层
 - 故障排除, 144
- 物理索引
 - 术语定义, 1117
- 物理限制
 - SQL Anywhere, 614
- X**
- X.509 证书
 - 创建, 705
 - 查看, 708
- XML 文件格式
 - Interactive SQL 输出, 680
- xp_cmdshell 系统过程
 - 安全功能, 1005
- xp_sendmail 系统过程
 - 安全功能, 1005
- xp_srvmon_count_unsubmitted_crash_reports 过程
 - 监控器, 950
- xp_startmail 系统过程
 - 安全功能, 1005
- xp_startsmtp 系统过程
 - 安全功能, 1005
- xp_stopmail 系统过程
 - 安全功能, 1005
- xp_stopsmtmp 系统过程
 - 安全功能, 1005
- XPathCompiles 属性
 - 数据库属性说明, 601
- X 窗口服务器
 - 在 Linux 上显示 SQL Anywhere UI, 219
- 系统 dbspace
 - 关于, 13
- 系统表
 - preserve_source_format, 534
 - prevent_article_pkey_update, 535
 - 术语定义, 1117
- 源列, 534, 535
- 系统对象
 - 卸载, 804
 - 术语定义, 1117
- 系统故障
 - 关于, 846
 - 恢复, 846
- 系统事件
 - BackupEnd, 868
 - Connect, 868
 - ConnectFailed, 868
 - DatabaseStart, 868
 - DBDiskSpace, 868
 - Deadlock, 868
 - Disconnect, 868
 - GlobalAutoIncrement, 868
 - GrowDB, 868
 - GrowLog, 868
 - GrowTemp, 868
 - LogDiskSpace, 868
 - MirrorFailover, 869
 - MirrorServerDisconnect, 869
 - RAISERROR, 869
 - ServerIdle, 869
 - TempDiskSpace, 868
 - 关于, 868
 - 内部, 873
 - 定义, 864
 - 数据库镜像, 901
- 系统视图
 - 术语定义, 1117
 - 权限, 458
 - 用户和组, 458
- 系统信息文件
 - 使用数据源实用程序 [dbdsn], 713
 - 关于, 95
 - 在 DSN 连接参数中指定, 262
 - 在 Mac OS X 上创建数据源, 93
 - 存储加密口令, 265
- 系统要求
 - Veritas 聚簇服务器代理, 905
- 下部代码页
 - 关于, 385
- 下载
 - 术语定义, 1117
- 显式选择性估计
 - user_estimates 选项, 559

- 陷阱
 - 与 SQL Anywhere SNMP Extension Agent 结合使用, 965
 - 关于, 957
 - 动态陷阱, 966
- 限定名
 - 数据库对象, 450
 - 表, 447
- 限制
 - SQL Anywhere, 614
 - SQL Anywhere 在 Windows Mobile 5 for Smartphone 上, 311
 - 临时文件, 552, 614
 - 临时表, 614
 - 事件, 614
 - 列, 614
 - 加密密钥, 1023
 - 口令, 431, 614
 - 在 Windows Mobile 上运行 SQL Anywhere, 336
 - 备份期间, 824
 - 恢复期间, 824
 - 数据库, 614
 - 数据库名, 614
 - 数据库服务器名, 614
 - 标识符, 614
 - 监控器, 914
 - 索引, 614
 - 表, 614
 - 语句, 614
 - 高速缓存大小, 614
- 线程
 - SQL Anywhere 中的线程, 47
 - Unix 行为, 47
 - Windows, 48
 - 关于, 47
 - 多个处理器, 185
 - 执行, 182
 - 控制行为, 48
- 线程应用程序
 - 用于 Unix 的 dbping_r, 757
- 相关名
 - 术语定义, 1117
- 详细模式
 - 卸载 [dbunload] 实用程序, 795
 - 日志传送管理器 [dbltn] 实用程序, 748
- 项目
 - 术语定义, 1117
- 向导
 - Windows Mobile 上不支持 Sybase Central 向导, 339
- 消除
 - 数据库, 32
 - 数据库从 Windows Mobile 设备, 328
- 消除实用程序 [dberase]
 - 使用, 32
 - 语法, 722
 - 退出代码, 723
- 消除数据库
 - Sybase Central, 32
 - 从 Windows Mobile 设备, 328
 - 关于, 722
- 消除数据库向导
 - Windows Mobile 上不支持, 339
 - 使用, 32
- 消息
 - 语言资源库, 384
- 消息存储库
 - 术语定义, 1117
- 消息代理
 - 事务日志管理, 847
- 消息类型
 - 术语定义, 1117
- 消息链接参数
 - SQL Remote external_remote_options, 503
- 消息日志
 - 关于, 40
 - 术语定义, 1118
- 消息系统
 - 术语定义, 1118
- 小数精度
 - 数据库选项, 533
- 校验实用程序 [dbvalid]
 - 语法, 810
- 校验数据库向导
 - 校验表, 860
- 协议
 - 使用 TCP/IP 的数据库服务器, 284
 - 关于, 133
 - 受支持的, 134
 - 故障排除, 143
 - 选择, 51
 - 选项, 51
- 协议选项
 - (参见 连接参数)

- 使用 HTTP 的数据库服务器, 284
- 使用 HTTPS 的数据库服务器, 284
- 列表, 284
- 布尔值, 284
- 按字母顺序排序的列表, 284
- 数据库服务器, 247
- 卸载
 - 安全, 1018
 - 术语定义, 1118
- 卸载实用程序 [dbunload]
 - dbspace 文件名, 795
 - 语法, 795
 - 退出代码, 805
- 卸载数据
 - 将计算列重新计算, 795
 - 安全, 1018
 - 指定字符集, 252
- 卸载数据库
 - 将计算列重新计算, 795
 - 卸载实用程序 [dbunload], 795
- 卸载数据库向导
 - Windows Mobile 上不支持, 339
- 新闻组
 - 技术支持, xvii
- 信息实用程序 [dbinfo]
 - 语法, 728
 - 退出代码, 728
- 行
 - 使用 Interactive SQL 删除, 656
 - 使用 Interactive SQL 添加, 655
 - 在 Interactive SQL 中复制, 656
 - 在 Interactive SQL 中插入, 655
 - 在 Interactive SQL 中编辑值, 654
- 行号
 - SQL 语句窗格, 639
- 行级触发器
 - 术语定义, 1101
- 行计数
 - 启用, 542
- 性能
 - LTM, 1087
 - OLAP 查询, 530
 - PowerBuilder DataWindow, 528
 - TCP/IP, 136
 - 主键, 16
 - 事务日志优点, 14
 - 事务日志大小, 16
 - 事务日志镜像, 15
 - 加密数据库, 1024
 - 压缩, 141
 - 改善, 862
 - 数据库镜像, 902
 - 服务器选项, 38, 44
 - 测试嵌入式 SQL 连接, 130
 - 磁盘碎片整理, 26
 - 结果集, 528
 - 表加密的影响, 1026
 - 设置优先级, 535
 - 预取, 534
 - 高速缓存大小, 158, 159, 161, 162
- 性能统计
 - 术语定义, 1118
- 性能统计信息
 - 禁用收集, 190
- 休眠模式
 - Windows Mobile, 310
- 修复
 - 监控器资源, 940
- 修改根登录策略
 - 关于, 417
- 许可
 - 个人服务器与网络服务器之间的差别, 36
 - 使用服务器授权 [dblic] 实用程序进行添加, 766
 - 影响线程, 48
- 许可类型
 - 服务器授权 [dblic] 实用程序, 766
- 许可实用程序 (见 服务器授权实用程序 [dblic])
- 许可文件
 - 关于, 767
- 选项
 - (参见 数据库选项)
 - (参见 选项 [Ultralite])
 - ASE 兼容性选项, 473
 - createcert, 705
 - dbisqlc 实用程序, 720
 - Interactive SQL [dbisql] 实用程序, 740
 - Interactive SQL 选项, 667
 - isql_allow_read_client_file, 673
 - isql_print_result_set, 677
 - 校验 [dbvalid] 实用程序, 810
 - Linux 服务 [dbsvc] 实用程序, 769
 - MobiLink 证书创建 [createcert], 705
 - MobiLink 证书查看器 [viewcert] 实用程序, 708
 - Open Client, 1067

ping [dbping] 实用程序, 757
PUBLIC 选项, 463
SQL Anywhere MIB 中的数据库选项, 987
SQL Anywhere MobiLink 客户端设置, 475
SQL Anywhere 控制台 [dbconsole] 实用程序, 780
SQL Anywhere 脚本执行 [dbrunsql] 实用程序, 761
SQL Remote 复制选项, 476
sr_date_format, 547
Transact-SQL 兼容性选项, 473
viewcert 实用程序, 708
Windows 服务 [dbsvc] 实用程序, 773
事务日志 [dblog] 实用程序, 792
信息 [dbinfo] 实用程序, 728
停止服务器 [dbstop] 实用程序, 783
关于, 462
分类, 466
初始化 [dbinit] 实用程序, 729
初始设置, 465
删除设置, 466
区分大小写, 463
升级 [dbupgrad] 实用程序, 808
卸载 [dbunload] 实用程序, 795
后台启动服务器 [dbspawn] 实用程序, 781
启动设置, 1067
在 Interactive SQL 中设置, 667
备份 [dbbackup] 实用程序, 698
广播转发器 [dbns11] 实用程序, 703
归类定制, 396
按字母顺序排序的数据库选项列表, 466
支持 [dbsupport] 语法, 785
数据库服务器, 148
数据库选项的作用域和持续时间, 463
数据源 [dbdsn] 实用程序, 710
日志传送管理器实用程序 [dbltn] 实用程序, 748
服务器授权 [dblic] 实用程序, 766
服务器枚举 [dblocate] 实用程序, 763
查找值, 464
消除 [dberase] 实用程序, 722
监控设置, 465
直方图 [dbhist] 实用程序, 726
翻译日志文件 [dbtran] 实用程序, 753
设置临时, 463
设置数据库选项, 462
设置用户和组选项, 433
语言选择 [dblang] 实用程序, 745

选项监视项目列表
关于, 465
选择归类
关于, 395
注意事项, 395
选择数据库镜像模式
关于, 883
选择性估计
user_estimates 选项, 559

Y

压缩
加密的数据库文件, 1019
包, 200
性能, 141
验证
数据库, 495
连接, 491
验证签名
关于, 70
验证数据库
关于, 71
验证应用程序
关于, 71
要求
Veritas 聚簇服务器代理, 905
页
事务日志, 14
页面
在数据库文件中显示使用情况, 728
页面大小
允许的最大值, 183
数据库, 729
选择, 729
选项, 46
页面模式
数据库镜像, 884
页面使用
信息 [dbinfo] 实用程序, 728
业务规则
术语定义, 1118
依赖性
Linux 服务的设置, 770
服务, 66
管理服务依赖性, 67
设置, 774
疑难解答

HTTP 客户端, 229
Kerberos 连接, 113
ODBC, 757
加密数据库性能, 1024
备份, 17
数据库服务器, 229
数据库服务器请求记录, 230, 232
数据库连接, 124
新闻组, xvii
标识客户端应用程序, 249
监控器, 952
结果, 387
连接, 124, 144, 757, 763
已提升的操作代理
 Vista, 38
以太网
 关于, 144
异步 I/O
 禁止在 Linux 上使用, 215
异步过程
 关于, 1086
 复制服务器, 1071
 用户 ID, 750
异步满页模式
 数据库镜像, 884
异步模式
 数据库镜像, 884
引号
 在连接字符串内使用, 81
引擎
 (参见 数据库服务器)
 (参见 服务器)
引用对象
 术语定义, 1118
应用程序
 SQL Anywhere OEM 版, 70
 SQL Anywhere Web 版应用程序, 75
应用程序分析
 在 Windows Mobile 上受限制, 336
应用程序分析模式
 关于, 631
硬件镜像
 事务日志, 15
映像备份
 重命名原始事务日志, 849
 使用备份 [dbbackup] 实用程序运行, 698
 使用备份实用程序 [dbbackup] 创建, 698
 关于, 823
 定义的, 823
 并行, 855
 恢复, 834
 接收来自 dbbackup 的错误消息, 698
用户
 REMOTE 权限, 439
 从登录策略中删除, 418
 从组中删除, 445
 分配登录策略, 418
 创建, 430
 创建 Kerberos 登录, 111
 删除, 440
 删除 Kerberos 登录, 111
 删除集成登录, 101
 授予集成登录, 99
 权限, 430
 权限冲突, 456
 添加, 430
 添加到组, 444
 登录策略, 416
 监控器创建, 942
 监控器删除, 944
 监控器发送电子邮件, 943
 监控器只读用户, 942
 监控器安全, 944
 监控器操作员, 942
 监控器管理员, 942
 监控器管理员用户, 942
 监控器类型, 942
 监控器编辑, 943
 监控器缺省用户, 942
 管理, 430
 设置选项, 433
 连接的用户, 442
 限制临时空间, 522
用户 ID
 DBA 特权, 425
 Guest, 105
 PUBLIC 选项, 463
 关于, 422
 列表, 458
 安全功能, 1004
 安全提示, 1005
 最大长度, 614
 登录策略, 416
 管理, 415

- 设置单个用户 ID, 430
- 用户标识
 - Windows Mobile 上的 SQL Anywhere 数据库, 1004
- 用户定义数据类型
 - 术语定义, 1119
- 用户估计
 - 替换, 559
- 用户名
 - 服务器授权 [dblic] 实用程序, 766
- 用户授权
 - Windows Mobile 上的 SQL Anywhere 数据库, 1004
- 用户提供的选择性估计
 - user_estimates 选项, 559
- 用户帐户控制
 - Vista, 38
 - 在 Vista 上运行 SQL Anywhere, 38
- 优化程序
 - 捕获最近计划, 230
 - 控制用于查找访问计划而消耗的资源, 529
 - 跳过, 530
- 优先级
 - 进程, 178
- 游标
 - ansi_close_cursors_on_rollback 选项, 481
 - close_on_endtrans 选项, 489
 - max_cursor_count 选项, 517
 - 和事务, 489
 - 数据库选项, 463
 - 术语定义, 1119
 - 连接限制, 457
- 游标结果集
 - 术语定义, 1119
- 游标位置
 - 术语定义, 1119
- 语法
 - 重建实用程序, 760
 - dbbackup 实用程序, 698
 - dbconsole 实用程序, 780
 - dbdsn 实用程序, 710
 - dberase 实用程序, 722
 - dbfhide 实用程序, 724
 - dbhist 实用程序, 726
 - dbinfo 实用程序, 728
 - dbinit 实用程序, 729
 - dbisql 实用程序, 740
 - dbisqlc 实用程序, 720
 - dblang 实用程序, 745
 - dblic 实用程序, 766
 - dblocate 实用程序, 763
 - dblog 实用程序, 792
 - dbltn 实用程序, 748
 - dbns11 实用程序, 703
 - dbping 实用程序, 757
 - dbrunsql 实用程序, 761
 - dbspawn 实用程序, 781
 - dbstop 实用程序, 783
 - dbsupport 实用程序, 785
 - dbsvc 实用程序 (Linux), 769
 - dbsvc 实用程序 (Windows), 773
 - dbtran 实用程序, 753
 - dbunload 实用程序, 795
 - dbupgrad 实用程序, 808
 - dbvalid 实用程序, 810
 - MobiLink 证书创建 [createcert], 705
 - MobiLink 证书查看器 [viewcert], 708
 - 密钥对生成器 [createkey] 语法, 744
 - 访问数据库属性, 601
 - 访问数据库服务器属性, 588
 - 访问连接属性, 566
 - 证书创建 [createcert], 705
 - 证书查看器 [viewcert] 实用程序, 708
- 语法错误
 - 连接, 503
- 语句
 - Windows Mobile 上不支持的语句, 337
 - 在 Interactive SQL 中记录, 647
 - 实用程序数据库, 28
 - 客户端高速缓存, 516
 - 记录, 42
 - 限制, 614
- 语句级触发器
 - 术语定义, 1119
- 语言
 - SQL Anywhere 的本地化版本, 378
 - 创建数据库, 401
 - 区分大小写, 388
 - 区域设置, 389
 - 土耳其语, 411
 - 客户端/服务器计算中的问题, 384
 - 指定, 355
 - 注册表设置, 374
 - 确定 CHAR 归类所支持的语言, 399

- 确定数据库服务器所使用的语言, 399
 - 语言选择 [dblang] 实用程序, 745
 - 软件和文档, 378
 - 非英语数据库, 378
 - 语言 DLL
 - 定位, 370
 - 注册表设置, 745
 - 语言标签
 - 值列表, 389
 - 语言代码
 - 语言 [dblang] 实用程序, 745
 - 语言实用程序 (见 语言选择实用程序)
 - 语言选择实用程序 [dblang]
 - 语法, 745
 - 退出代码, 746
 - 语言支持
 - 关于, 378
 - 多字节字符集, 392
 - 概述, 378
 - 语言资源库
 - 注册表设置, 745
 - 消息文件, 384
 - 域
 - 术语定义, 1119
 - 预定事件
 - 数据库镜像, 882
 - 预定义 dbspace
 - 关于, 13
 - 预订
 - 为复制服务器创建, 1080
 - 术语定义, 1120
 - 预计
 - user_estimates 选项, 559
 - 预取选项
 - DisableMultiRowFetch 连接参数, 263
 - 预准备语句
 - 连接限制, 457
 - 元数据
 - 术语定义, 1120
 - 元数据表
 - SQL Anywhere MIB, 969
 - 原子事务
 - 术语定义, 1120
 - 源代码控制
 - 与 Interactive SQL 集成, 658
 - 从 Interactive SQL 打开源代码控制项目, 660
 - 从 Interactive SQL 签入文件, 662
 - 从 Interactive SQL 签出文件, 661
 - 可从 Interactive SQL 中使用的操作, 662
 - 在 Interactive SQL 中配置, 658
 - 源代码控制操作列表
 - 使用, 659
 - 源代码控制项目
 - 从 Interactive SQL 打开, 660
 - 远程 ID
 - 术语定义, 1120
 - 远程数据访问
 - cis_option 选项, 488
 - cis_rowset_size 选项, 489
 - Windows Mobile 上不支持, 336
 - 远程数据库
 - 术语定义, 1120
 - 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
 - 术语定义, 1121
 - 运行
 - Interactive SQL 命令, 641
 - 运行服务器
 - 关于, 880
 - 运行数据库服务器
 - 概述, 35
 - 运营公司
 - 术语定义, 1121
- ## Z
- 在 Windows Mobile 上重建数据库
 - 关于, 326
 - 资源调控器
 - 定义的, 457
 - 游标, 517
 - 语句, 520
 - 脏页
 - 关于, 17
 - 增量备份
 - 关于, 821
 - 备份 [dbbackup] 实用程序, 827
 - 术语定义, 1121
 - 找不到数据库服务器
 - 搜索服务器, 126
 - 诊断目录

- SADIAGDIR 环境变量, 353
- 征用
 - 分布式事务, 214
- 争用
 - 术语定义, 1121
- 正则表达式
 - 术语定义, 1121
- 证书
 - certificate_company 协议选项, 287
 - certificate_name 协议选项, 288
 - certificate_unit 协议选项, 289
 - trusted_certificates 协议选项, 307
 - 传送层安全中的数字证书, 1036
 - 创建 ECC, 705
 - 创建 RSA, 705
 - 查看, 708
- 证书颁发机构
 - 传送层安全, 1040
- 证书查看器实用程序 [viewcert]
 - 语法, 708
- 证书撤销列表
 - 查看, 708
- 证书创建实用程序 [createcert]
 - 语法, 705
- 证书链
 - 传送层安全, 1037
- 证书请求
 - 查看, 708
- 证书实用程序
 - 传送层安全, 1054
- 支持
 - 新闻组, xvii
- 支持的平台
 - Kerberos, 107
 - SQL Anywhere SNMP Extension Agent, 956
- 支持实用程序 [dbsupport]
 - 使用, 76
 - 语法, 785
- 直方图
 - 使用 dbhist 查看, 726
 - 术语定义, 1121
- 直方图实用程序 [dbhist]
 - 语法, 726
 - 退出代码, 727
- 直接行处理
 - 术语定义, 1121
- 执行
 - Interactive SQL 中的命令, 641
 - Interactive SQL 中的多条语句, 643
 - 事件处理程序, 874
 - 命令文件, 643
- 执行 SQL 语句
 - Interactive SQL, 641, 642
- 执行计划
 - 打印, 653
- 执行线程
 - 数量, 182
- 执行语句
 - 验证语句, 72
- 值
 - 在 Interactive SQL 中编辑, 654
- 指定归类
 - 创建数据库, 401
- 指派登录策略
 - 关于, 418
- 只读
 - sm 选项, 241
 - 数据库, 46, 204, 240
 - 镜像数据库的访问, 898
- 只读用户
 - 监控器, 942
 - 监控器要求登录, 944
- 致命错误
 - uf 服务器选项, 217
 - 报告, 76
- 智能手机
 - SQL Anywhere 服务器的限制, 311
- 终端服务
 - 共享内存连接, 52
- 仲裁服务器
 - 停止, 900
 - 提供数据库名, 222
 - 提供连接字符串, 222
 - 数据库镜像概述, 880
 - 数据库镜像系统中的角色, 883
- 主表
 - 术语定义, 1122
- 主服务器
 - 从故障恢复, 900
 - 停止, 900
 - 强制故障转移, 899
 - 数据库镜像概述, 880
 - 确定, 897
- 主键

- 事务日志, 16
- 术语定义, 1122
- 主键约束
 - 术语定义, 1122
- 主题
 - 图标, xv
- 主站点
 - 使用 LTM, 1072
 - 创建, 1073
 - 复制服务器, 1071
 - 添加复制服务器信息, 1075
- 助手
 - 连接助手, 86
- 注册表
 - language 设置, 374
 - location 设置, 374
 - Sybase Central, 374
 - Windows Mobile, 374
 - Windows 服务, 373
 - 修改, 344
 - 关于, 373
 - 工具位置设置, 374
 - 环境变量, 344
 - 设置 SQLREMOTE 环境变量, 364
 - 语言选择 [dblang] 实用程序, 745
- 注释
 - 审计, 1015
- 注销
 - 保持服务器运行, 216
- 专用密钥
 - 查看, 708
- 转换
 - PKI 对象编码, 708
- 转义字符
 - 创建数据库, 21
 - 卸载 [dbunload] 实用程序, 795
- 装载
 - READCLIENTFILE 特权, 426
- 装载数据
 - 安全, 1018
- 状态
 - 监控器, 928
- 状态文件 (厶 状态信息文件)
- 状态信息文件
 - 在确定主服务器中的角色, 897
 - 数据库镜像, 885
- 资源
 - 监控器, 927
 - 子查询
 - 术语定义, 1122
 - 子目录
 - Windows Mobile, 368
 - 自动备份
 - 关于, 844
 - 自动恢复
 - 关于, 817
 - 自动启动数据库
 - Windows Vista, 264
 - 连接, 118
 - 自动完成
 - 管理任务, 864
 - 自签名证书
 - 为传送层安全创建, 1036
 - 传送层安全, 1036
 - 自然连接
 - 术语定义, 1110
 - 自由访问控制
 - 嵌套视图和表的规则, 454
 - 字符
 - 使用归类进行排序, 392
 - 字符 V 和 W
 - 在瑞典语 UCA 归类中识别, 398
 - 字符编码
 - 定义, 384
 - 字符串
 - 和主机变量, 480
 - 最大大小, 614
 - 术语定义, 1122
 - 字符集
 - ASE 标签, 404
 - IANA 标签, 404
 - IANA 标签列表, 408
 - ICU 标签, 404
 - Java 标签, 404
 - LTM, 1089
 - MIME 标签, 404
 - Open Client/Open Server 归类, 1088
 - Unicode, 392
 - Unix 缺省值, 391
 - Windows, 386
 - Windows 缺省, 391
 - 关于, 378
 - 初始化 [dbinit] 实用程序, 729
 - 单字节, 385

- 卸载数据, 252
- 可变宽度, 385
- 固定宽度, 385
- 土耳其语数据库, 411
- 在 SQL Anywhere 中, 386
- 复制服务器, 1083
- 多字节, 385
- 多字节归类, 392
- 定义, 384
- 应用程序, 391
- 建议在 Unix 平台上使用, 409
- 建议在 Windows 平台上使用, 408
- 所提供的 CHAR 编码列表, 404
- 指定, 352
- 替代编码, 404
- 服务器, 391
- 术语定义, 1122
- 标签, 408
- 确定 CHAR 字符集, 399
- 确定 NCHAR 字符集, 399
- 确定是否支持字符集, 404
- 编码, 378
- 转换, 386
- 连接参数, 252
- 字符集注意事项
 - LTM, 1088
- 字符集转换
 - ICU, 386
 - SQL 语句, 387
 - 关于, 386
 - 客户端/服务器, 387
- 字符替代
 - on_charset_conversion_failure 选项, 527
- 字体
 - 代码编辑器的设置, 626
- 组
 - PUBLIC, 448
 - REMOTE 权限, 439
 - SYS, 448
 - 关于, 422
 - 创建, 444
 - 删除, 448
 - 删除集成登录, 101
 - 成员资格, 444
 - 授予集成登录, 99
 - 撤消成员资格, 445
 - 无法继承登录策略, 416
 - 服务, 66
 - 权限, 429, 446
 - 权限冲突, 456
 - 没有口令, 447
 - 添加, 444
 - 添加用户, 444
 - 特权, 446
 - 离开, 445
 - 管理, 443
 - 设置依赖性, 774
 - 设置选项, 433
 - 限制临时空间, 522
- 组合
 - Interactive SQL 中的多条语句, 643
- 组依赖性
 - 设置, 774
- 组织
 - certificate_company 协议选项, 287
- 组织单位
 - certificate_unit 协议选项, 289
 - 在 MobiLink 传送层安全中进行校验, 1050
- 最大
 - 数据库大小, 614
 - 数据库文件大小, 614
- 最少列定义
 - 复制服务器, 1085
- 作为服务登录的特权
 - Linux 服务 [dbsvc] 实用程序, 770
 - Windows 服务 [dbsvc] 实用程序, 774