SYBASE®
An SAP® Company

安全性管理指南

Adaptive Server® Enterprise

15.7

文档 ID：DC01812-01-1570-01

最后修订日期：2011 年 9 月

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

目录

第 1 章

安全性简介

.

安全性简介

信息可能是公司的最重要资产。信息与其它资产一样，也需要保护。系统管理员应确定如何以最佳方式保护公司数据库中包含的信息，以及谁可访问该信息。单个数据库服务器需要功能强大而灵活的安全性支持。

用户和用户所访问的数据可能位于世界各地，由非受托网络连接。确保这种环境中的敏感数据和事务的机密性和完整性至关重要。

只有需要信息的人在需要时可以获得信息，信息才是有用的。由于商业关系复杂多变，因此只允许经过授权的用户访问信息至关重要。

什么是 “信息安全性”？

在考虑企业的安全性时，应遵循以下一些一般原则：

- 敏感的信息应保持机密性 — 请确定哪些用户应能访问哪些信息。
- 系统应强制实施完整性保护 — 服务器应强制执行规则和约束以确保信息准确和完整。
- 信息应保持可用 — 即使已实施了所有安全保护措施，任何需要访问信息的人在需要信息时也应该能够访问到此信息。

明确公司要保护的内容，以及外部有关各方对公司有何要求：

- 明确信息资产以及当它们易受攻击或受到损害时会有哪些安全风险。
- 明确并了解适用于公司和信息资产的所有法律、法令、规章以及合约协议。
- 明确公司的业务流程及其对信息资产的要求，以兼顾实际业务需要和安全风险。

安全要求会随着时间的推移而改变。请定期重新评估安全要求，确保它们仍能体现出公司的需要。

接着，设置一系列满足公司安全性目标的控制和策略，这样做可以建立一个信息安全策略文档，该文档阐明了为信息安全性所做的决策。

Adaptive Server® 包含了一组安全性功能，可帮助您强制实施公司的安全策略。有关 Adaptive Server 中安全性功能的详细信息，请参见第 2 章“Adaptive Server 中的安全性管理快速入门”。

信息安全性标准

已依照公共标准评估和验证方案的条款对 Adaptive Server 进行了评估和验证。 Adaptive Server 还使用 FIPS 140-2 认证模块实现加密功能。

本节将介绍这些认证。

公共标准配置评估

信息技术安全性评估公共标准是计算机安全性认证的一项国际标准 (ISO/IEC 15408)。公共标准由加拿大、法国、德国、荷兰、英国和美国的政府制定。

Adaptive Server 15.0.1 版于 2007 年 9 月完成了公共标准验证。“已评估的配置”包括具有安全性和目录服务选项的 Adaptive Server 15.0.1 版。Adaptive Server 安全性评估是依照公共标准评估和验证方案 (CCEVS) 的过程和方案来实施的。在信息技术安全性评估的公共标准 2.3 版和 2005 年 8 月生效的国际解释中介绍了判定 Adaptive Server Enterprise 所依据的标准。如果按照 Supplement for Installing Adaptive Server for Common Criteria Configuration （《安装用于公共标准配置的 Adaptive Server 的补充说明》）中的说明来配置 Adaptive Server，则 Adaptive Server 符合 Sybase® Adaptive Server Enterprise 安全性目标 （1.5 版）中规定的所有安全性功能要求。

Adaptive Server 支持八种安全性功能：

- 加密支持 — Adaptive Server 支持列级的透明数据加密。SQL 语句和扩展提供了安全的密钥管理。
- 安全性审计 — 检查访问、鉴定尝试和管理员功能的审计机制。该安全性审计功能记录了日期、时间、负责人和在审计追踪中描述事件的其它详细信息。
- 用户数据保护 — Adaptive Server 对相应的数据库对象执行自由选择访问控制策略：数据库、表、视图、存储过程和加密密钥。
- 识别和鉴定 — 除了基础操作系统机制以外，Adaptive Server 还提供它自己的识别和鉴定机制。
- 安全性管理 — 允许您管理用户和相关特权、访问权限的功能以及其它安全性功能 （例如审计追踪）。这些功能受到自由选择访问控制策略规则 （包括角色限制）的限制。
- 保护 TOE 安全性功能 (TSF) — Adpative Server 将其环境与其用户的环境分离并使用操作系统机制，可确保 Adaptive Server 使用的内存和文件具有适当的访问设置。 Adaptive Server 通过定义明确的界面与用户进行交互，旨在确保强制执行其安全性策略。
- 资源利用 — Adaptive Server 提供资源限制，以防止查询和事务独占服务器资源。
- 评估目标 (TOE) 访问 — Adaptive Server 允许经过授权的管理员构造登录触发器，以限制登录到特定数量的会话，并基于时间限制访问。经过授权的管理员还可以基于用户身份限制访问。

FIPS 140-2 验证的加密模块

SSL 是一种用于保护通过 Internet 传输敏感信息（例如信用卡号、股票交易和银行交易）的标准。Adaptive Server 的 SSL 使用 Certicom Security Builder GSE（一种 FIPS 140-2 级别 1 验证的加密模块。请参见 NIST 网站 http://csrc.nist.gov 上的验证证书 #542（日期为 2005 年 6 月 2 日）。

如果启用了 FIPS login password encryption 配置参数， FIPS 140-2 认证的 Certicom Security Builder GSE 还用于对传输的登录包中、内存中和磁盘上的登录口令进行加密。

注释 需要安全和目录服务许可证才能使用 SSL 和启用 FIPS login password encryption 参数。如果未启用该参数，则使用 OpenSSL 安全提供程序来执行登录口令加密。

Adaptive Server 加密列功能依赖于对称密钥密码术，并使用相同的 FIPS 140-2 验证的加密模块作为 SSL。请参见《加密列用户指南》。

注释 您必须有加密列许可证才能使用 Adaptive Server 加密列功能。

第 2 章

Adaptive Server 中的安全性管理快速入门

安全性管理的一般过程

“执行主要任务以安全地管理 Adaptive Server”介绍了以安全方式管理 Adaptive Server 所需执行的主要任务，并介绍了包含用于执行每个任务的说明的文档。

❖ 执行主要任务以安全地管理 Adaptive Server

1 安装 Adaptive Server，包括审计 — 包括准备安装、从发行介质中装载文件、执行实际的安装，以及管理所需的物理资源。请参见所用平台的安装文档和第 8 章 “审计”。

2 建立安全的管理环境 — 设置系统管理员和系统安全员、创建登录配置文件，以及建立口令和登录策略。请参见第 3 章 “管理 Adaptive Server 登录名和数据库用户”。

3 设置登录名、数据库用户和角色 — 向服务器中添加用户登录名并将登录配置文件分配给它们。创建用户定义的角色、定义角色层次和角色的互斥性，以及为登录名指派角色。向数据库添加用户。请参见第 3 章 “管理 Adaptive Server 登录名和数据库用户”。

4 管理用户、组和角色的权限 — 授予和撤消某些 SQL 命令、执行某些系统过程以及访问数据库、表、特定表列和视图的权限。创建访问规则以强制实施精细访问控制。请参见第 6 章 “管理用户权限”。

5 在数据库中配置加密以对表中的敏感数据进行加密。对敏感数据进行加密 — 配置 Adaptive Server 以使用列级加密，决定要对哪些列数据进行加密，执行一次性密钥创建操作，并使用 alter table 执行初始数据加密。请参见 《加密列用户指南》。

6 建立对数据的完整性控制 — 添加检查约束、域角色和参照约束以验证传入数据。请参见 《Transact-SQL 用户指南》和 《参考手册：命令》。

7 设置和维护审计 — 确定审计对象，审计 Adaptive Server 的使用情况以及使用审计追踪检测系统渗透和资源误用情况。请参见第 8 章 “审计”和所用平台的 Adaptive Server 安装和配置文档。

8 设置安装以使用高级鉴定机制和网络安全性 — 配置服务器以使用诸如基于 LDAP、PAM 或 Kerberos 的用户鉴定、通过加密实现数据保密性以及数据完整性等服务。请参见第 4 章 “外部鉴定”和第 7 章 “数据的保密性”。

设置安全性的建议

以下内容介绍了登录名以及登录名与安全性之间的关系。

- 使用 “sa” 登录名 — 当您安装 Adaptive Server 时，将使用系统管理员和系统安全员角色配置一个名为 “sa” 的单一登录名，这意味着 “sa” 登录名对数据库中进行的操作具有无限制的控制。

  只在初始设置时使用 “sa” 登录名。不要允许几个用户使用 “sa” 帐户，而是要通过将特定角色指派给个别管理员来建立个人责任。

- 更改 “sa” 登录口令 — “sa” 登录名配置为最初使用 “NULL” 口令登录。在安装后，立即使用 alter login 命令更改口令。

  警告！ 登录到 Adaptive Server 时，不要使用 isql 命令的 -P 选项来指定口令，因为其他用户可能有机会看到该口令。

- 启用审计 — 在管理过程中尽早启用审计，以便记录由系统安全员和系统管理员执行的特权命令。可能还需要审计由具有其它特殊角色的成员执行的命令，例如操作员在转储和装载数据库时

- 指派登录名 — 指派与各自操作系统的登录名相同的 Adaptive Server 登录名。这将使登录到 Adaptive Server 更加容易，简化了服务器和操作系统登录帐号的管理，同时也使关联 Adaptive Server 生成的审计数据和操作系统生成的审计数据更加容易。

设置安全性的示例

此示例使用指派给表 2-1 所列出用户的特殊角色。

表 2-1：将为其指派角色的用户

名称	特权	操作系统登录名
Rajnish Smith	sso_role	rsmith
Catherine Macar-Swan	sa_role	cmacar
Soshi Ikedo	sa_role	sikedo
Julio Rozanski	oper_role	jrozan
Alan Johnson	dbo	ajohnson

表 2-2 根据表 2-1 所示的角色指派，显示了为 Adaptive Server 设置安全操作环境可能使用的命令序列。登录到操作系统后，使用初始 “sa” 帐号发出这些命令。

表 2-2：用于设置安全性的命令示例

命令	结果
• isql -Usa	以 “sa” 身份登录到 Adaptive Server。 sa_role 和 sso_role 均处于活动状态。
• sp_audit “security”, “all”, “all”, “on”‘ • sp_audit “all”, “sa_role”, “all”, “on” • sp_audit “all”, “sso_role”, “all”, “on”‘	为全服务器范围内与安全性相关的事件以及对激活了 sa_role 或 sso_role 的所有操作的审计设置审计选项。
• sp_configure “auditing”, 1	启用审计。

注释　在启用审计前，请为审计追踪设置一个阈值过程并确定如何处理 sybsecurity 中的事务日志。请参见第 8 章 “审计”。

• create login	添加登录名和口令。
• grant role	授予角色。
• use sybsecurity • sp_changedbowner rsmith	通过使系统安全员 Rajnish 成为数据库所有者，授予访问审计数据库 sybsecurity 的权限。未授予 Alan 任何系统定义角色。
sp_locklogin sa,"lock"	锁定 “sa” 登录名，以便没有人能够以 “sa” 身份登录。个人只能使用为他们配置的角色。

注释　在为个别用户授予 sa_role 和 sso_role 角色并且证实可成功使用之前，不要锁定 “sa” 登录名。

Adaptive Server 中的安全性功能

Adaptive Server 中的主要安全性功能有：

- 第 8 页的 “标识和鉴定” — 确保只有经过授权的用户才能登录到系统。除了基于口令的登录鉴定外， Adaptive Server 还支持使用 Kerberos、 LDAP 或 PAM 的外部鉴定。
- 第 9 页的 “自由选择访问控制” — 通常使用 grant 和 revoke 命令提供访问控制，使对象所有者能够限制对对象的访问。此类控制取决于对象所有者的选择。
- 第 10 页的 “角色分离” — 允许管理员将特权角色授予指定用户，以便只有指定的用户才能执行某些任务。 Adaptive Server 包含一些预定义角色 （称为 “系统角色”），例如系统管理员和系统安全员。此外， Adaptive Server 允许系统安全员定义其它角色 （称为 “用户定义角色”）。
- 第 11 页的 “责任审计” — 提供了审计事件 （例如登录、注销、服务器启动操作、远程过程调用、对数据库对象的访问，以及由特定用户或具有特定活动角色的用户执行的所有操作）的能力。 Adaptive Server 还提供了单个选项，用于审计全服务器范围内与安全性相关的一组事件。
- 第 12 页的 “数据的保密性” — 对客户端/服务器通信使用加密来维持数据的保密性，可用 Kerberos 或 SSL 实现。列级加密可保持数据库中所存储数据的保密性。不活动数据的保密性是通过口令保护的数据库备份保持的。

标识和鉴定

Adaptive Server 使用服务器用户标识 (SUID) 来唯一地标识具有登录帐户名的用户。此标识链接到每个数据库中的特定用户标识 (UID)。访问控制使用此标识确定是否允许具有此 SUID 的用户访问对象。鉴定会检验某个用户实际上是否是其所声称的用户。 Adaptive Server 允许同时使用内部和外部鉴定机制。

标识和鉴定控制将在第 3 章 “管理 Adaptive Server 登录名和数据库用户” 中讨论。此外，还请参见第 178 页的 “使用代理授权” 和 《系统管理指南第一卷》中的第 7 章 “管理远程服务器”。

外部鉴定

通过使用中央存储库鉴定登录名，通常可以增强大型异构应用程序的安全性。 Adaptive Server 支持多种外部鉴定方法：

- Kerberos — 在包括 Kerberos 基本结构的企业环境中，提供一种集中且安全的鉴定机制。使用一台名为密钥分发中心的第三方受托服务器进行鉴定，以同时检验客户端和服务器。
- LDAP 用户鉴定 — 轻量目录访问协议 (LDAP) 可根据用户的登录名和口令，提供一种集中式鉴定机制。
- PAM 用户鉴定 — 可插入鉴定模块 (PAM) 提供一种集中式鉴定机制，该鉴定机制为管理和运行时应用程序操作使用操作系统界面。

有关每个外部鉴定方法的详细信息，请参见第 4 章 “外部鉴定”。

管理远程服务器

《系统管理指南第一卷》的第 7 章 “管理远程服务器”中介绍了用于管理 Adaptive Server 之间的登录名和用户的内部机制。

自由选择访问控制

对象所有者可将他们拥有的对象访问权限授予其他用户。数据库所有者还可以授予其他用户将访问权限转交给其他用户的权限。通过 Adaptive Server 自由选择访问控制，可以使用 grant 命令授予用户、组和角色各种权限。使用 revoke 命令撤消这些权限。 grant 和 revoke 命令授予用户执行指定命令和访问指定表、过程、视图、加密密钥和列的权限。

某些命令可由任何用户随时使用，而不需要任何权限。其它命令则只能由具有某种身份的用户 （例如系统管理员）使用，并且不可转交。

是否能够指派可被授予或撤消的命令的权限由每个用户的身份 （系统管理员、系统安全员、数据库所有者或数据库对象所有者）以及是否授予特定用户可将该权限授予其他用户的权限来决定。

自由选择访问控制在第 6 章 “管理用户权限”中讨论。

行级访问控制

行级访问控制提供了一种功能强大而灵活的保护数据（直到行级别）的方式。管理员定义的访问规则基于单个数据元素的值，而服务器会透明地强制执行这些规则。管理员定义访问规则后，只要通过应用程序、即席查询、存储过程、视图等查询受影响的数据，就会自动调用该规则。

使用基于规则的访问控制可简化 Adaptive Server 安装的安全性管理和应用程序开发过程，因为是服务器而不是应用程序强制执行安全性。利用以下功能，可以实现行级访问控制：

- 访问规则
- 应用程序环境功能
- 登录触发器
- 域完整性规则

请参见第 198 页的 “使用行级访问控制”。

角色分离

Adaptive Server 支持的角色允许您强制执行并维护个人责任。 Adaptive Server 提供了系统角色（例如系统管理员和系统安全员），以及由系统安全员创建的用户定义角色。

角色是权限集合，能让权限被授予者执行工作。角色为执行操作和管理任务的用户提供个人责任，并允许您审计操作并将操作归于这些用户。

角色层次

系统安全员可以定义角色层次，以便如果用户拥有一个角色，则将自动拥有层次中较低的角色。例如，“chief_financial_officer”角色可以包含“financial_analyst”和“salary_administrator”两个角色。首席财务官可以执行所有任务，并查看工资管理员和财务分析员可以查看的所有数据。

互斥性

可在成员资格级别或激活级别将角色定义为互斥。例如：

- 您可能不希望将 “payment_requestor”和 “payment_approver”这两个角色授予同一用户。
- 可以授予一个用户 “senior_auditor”和 “equipment_buyer”角色，但您可能不希望允许用户同时启用这两个角色。

可将系统角色以及用户定义角色定义为位于一个角色层次中，或定义为互斥。例如，您可能希望有一个 “super_user” 角色包含系统管理员角色、操作员角色和技术支持部门角色。另外，您可能希望将系统管理员和系统安全员角色定义为在成员资格级别互斥；即一个用户不能被同时授予这两个角色。

请参见第 139 页的 “为用户创建和指派角色”。

责任审计

Adaptive Server 包括全面的审计系统。审计系统包括：

- sybsecurity 数据库
- 用于管理审计的配置参数
- sp_audit （用于设置所有审计选项）
- sp_addauditrecord （用于将用户定义记录添加到审计追踪）

安装审计后，可以指定 Adaptive Server 用于审计追踪的审计表的数量。如果使用两个或多个表来存储审计追踪，可以建立一个可以平稳运行而且无须手动干涉又不会丢失记录的审计系统。

系统安全员管理审计系统，且只有系统安全员可以启动和停止审计、设置审计选项以及处理审计数据。作为系统安全员，您可以为诸如以下事件建立审计：

- 全服务器范围内的与安全性相关的事件
- 创建、删除和修改数据库对象
- 特定用户执行的所有操作或具有特定活动角色的用户执行的所有操作
- 授予或撤消数据库访问权限
- 导入或导出数据
- 登录和注销
- 与加密密钥相关的所有操作

审计功能在第 8 章 “审计”中讨论。

数据的保密性

Adaptive server 通过使用安全套接字层 (SSL) 标准或 Kerberos 来加密客户端/服务器通信，以便您可以保持数据的保密性。通过在数据库中使用列级加密和对脱机数据的备份进行加密，您可以保护数据的保密性。dump 和 load database 命令包括一个 *password* 参数，该参数允许您为数据库转储提供口令保护。

有关详细信息，请参见：

- SSL — 第 7 章 “数据的保密性”
- Kerberos — 第 4 章 “外部鉴定”
- 加密列 — 《加密列用户指南》
- 转储和装载 — 《参考手册：命令》和 《系统管理指南：第二卷》中的第 12 章 “备份和恢复用户数据库”

第 3 章

管理 Adaptive Server 登录名和数据库用户

登录名和登录配置文件简介

登录名定义用户用以访问 Adaptive Server 的名称和口令。当您执行 create login 时， Adaptive Server 将向 master.dbo.syslogins 中添加一行，为新用户指派唯一的系统用户 ID (suid)，并填入指定的属性信息。用户登录时， Adaptive Server 在 syslogins 中查找用户提供的名称和口令。password 列使用单向算法加密，因此不可读。

登录配置文件是由应用于一组登录帐户的属性组成的集合。这些属性定义登录特征，如缺省角色或与绑定到配置文件的每个登录名相关联的登录脚本。登录配置文件可节省系统安全管理员的时间，因为登录帐户的属性是在一个地方设置和管理的。

管理登录帐户

向 Adaptive Server 中添加新的登录帐户，向数据库中添加用户，以及授予用户使用命令和访问数据库对象的权限的职责由系统安全员、系统管理员和数据库所有者分担完成。

表 3-1 总结了用于创建和管理登录帐户的系统过程和命令。

表 3-1：在 Adaptive Server 中管理用户

任务	要求的角色	命令或过程	数据库、组或角色
创建登录帐户	系统安全员	create login	master 数据库
改变登录帐户	系统安全员 例外情况是用户可以更改自己的口令和全名。	alter login	master 数据库
删除登录帐户	系统安全员	drop login	master 数据库
创建组	数据库所有者或系统管理员	sp_addgroup	用户数据库
创建并指派角色	系统安全员	create role、grant role	master 数据库
向数据库中添加用户和指派组	数据库所有者或系统管理员	sp_adduser	用户数据库
为用户授予其他数据库用户的别名	数据库所有者或系统管理员	sp_addalias	用户数据库
为组、用户或角色授予创建或访问数据库对象以及运行命令的权限	数据库所有者、系统管理员、系统安全员或对象所有者	grant	用户数据库

创建登录帐户

以下步骤讲述如何为特定服务器创建登录帐户以及为用户管理权限。

1　系统安全员为新用户创建登录帐户。

2　系统管理员或数据库所有者向数据库中添加用户或为组分配用户。

3　系统安全员授予用户特定的角色。

4　系统管理员、数据库所有者或对象所有者授予用户或组对特定命令和数据库对象的特定权限。

使用 create login 命令可向 Adaptive Server 添加将新登录名。只有系统安全员才能执行 create login。

有关完整语法，请参见 《参考手册：命令》中的 create login。

创建登录时， syslogins 的 crdate 列设置为当前时间。

syslogins 中的 suid 列在 Adaptive Server 中唯一地标识每个用户。不论用户正在使用什么数据库，用户的 suid 都保持不变。总是将 suid 1 指派给安装 Adaptive Server 时创建的缺省 “sa” 帐户。其他用户的服务器用户 ID 是由 Adaptive Server 在每次执行 create login 时连续指派的整数。

有关选择口令的信息，请参见选择和创建口令。

下面的语句为用户 “maryd” 建立一个帐户，口令为 “100cents”，使用缺省数据库 (master)、缺省语言 (us_english)，且没有全名：

```
create login maryd with password "100cents"
```

由于口令以 1 开头，因此要用双引号引起来。

执行完此语句后，“maryd” 就可以登录到 Adaptive Server。除非特别授予该用户对 master 的访问权限，否则系统会自动将其作为 master 数据库的 “guest” 用户对待，只具有有限的权限。

下面的语句设置一个登录帐户（“omar_khayyam”）和口令（“rubaiyat”），并使 “pubs2” 成为此用户的缺省数据库：

```
create login omar_khayyam with password rubaiyat
default database pubs2
```

上次登录和管理不活动帐户

Adaptive Server 通过以下方式为用户帐户提供安全性：

- 跟踪创建日期。
- 记录帐户的上次登录时间。
- 确定哪些帐户由于不活动而过时并被锁定。
- 记录锁定帐户的原因、锁定帐户的时间，以及锁定帐户的用户的标识。

定义 stale period

stale period 是登录配置文件的属性，指示允许登录帐户在因不活动而被锁定前保持不活动状态的持续时间。如果登录配置文件 track lastlogin 属性未设置为 0，并且不让登录帐户因不活动而被锁定，则会在登录过程中或执行 sp_locklogin 过程中检查 syslogins.lastlogindate 和 syslogins.pwdate 字段以确定是否不活动。

如果登录帐户因不活动而被锁定，则 syslogins 中的 locksuid、 lockreason 和 lockdate 字段将设置如下：

lockreason 的值	locksuid 的值	帐户 lockreason 的说明
4	NULL	帐户由于不活动而被自动锁定。

如果设置了高可用性解决方案，则 syslogins.lastlogindate 和 syslogins.pwdate 会在两个服务器上同步。在一个服务器上锁定的登录帐户也会在协同服务器上锁定。

跟踪上次登录

可以通过登录配置文件的 track lastlogin 属性设置跟踪上次登录日期时间。

```
create login profile general_lp with track lastlogin
true authenticate with ASE
```

防止不活动帐户被锁定

可以使用 exempt inactive lock 子句将登录帐户设置为不因不活动而被锁定。

以下语句创建一个不因不活动而被锁定的登录帐户 “user33”。

```
create login user33 with password AT0u7gh9wd exempt
inactive lock true
```

登录的鉴定机制

支持的鉴定机制有：ASE、 LDAP、 PAM、 KERBEROS 和 ANY。

当使用 ANY 时， Adaptive Server 会检查有无定义的外部鉴定机制。如果定义了外部鉴定机制，Adaptive Server 会使用它，否则会使用 ASE 机制。

更改登录帐户

使用 alter login 可添加、删除或更改登录名的属性及其相应的值。 alter login 能让您：

- 添加或删除自动激活的角色
- 更改口令
- 更改登录配置文件关联
- 更改或添加全名
- 指定口令有效期和最小口令长度
- 指定最大尝试失败次数
- 指定鉴定机制
- 指定缺省语言和缺省数据库
- 调用登录脚本
- 不锁定不活动的登录帐户

系统管理员可以使用alter login 来设置口令长度和有效期、限制失败登录尝试次数、删除属性，以及指定登录脚本在用户登录时自动运行。

执行 alter login 更改缺省数据库后，用户将在下次登录时连接到新的缺省数据库。但 alter login 不会自动授予用户访问此数据库的权限。除非使用 sp_adduser、 sp_addalias 或 guest 用户机制对数据库所有者指派了访问权限，否则即使其缺省数据库已经更改，用户也将被连接到 master 数据库。

下面的示例将登录帐户 anna 的缺省数据库更改为 pubs2：

```
alter login anna modify default database pubs2
```

下面的示例将 claire 缺省语言更改为法文：

```
alter login claire modify default language french
```

删除登录帐户

命令 drop login 通过删除 master.dbo.syslogins 中的用户条目删除 Adaptive Server 用户登录名。

您无法删除是任何数据库中的用户的登录名；并且如果来自某一数据库的用户在该数据库中拥有任何对象或已将对对象的任何权限授予其他用户，您也无法删除该用户。

在创建下一登录帐户时，可以重用已删除的登录帐户的服务器用户 ID (suid)。这种情况只会在被删除的登录名持有 syslogins 中最大 suid 时才出现；但如果未审计 drop login 的执行情况，则它可能危及责任的安全。

您不能删除剩余的最后一个系统安全员的登录帐户或系统管理员的登录帐户。

with override 子句删除登录名，即使有无法检查其有无登录引用的不可用数据库也是如此。

下面的示例删除登录帐户 mikeb 和 rchin。

```
drop login mikeb, rchin
```

有关完整的 drop login 语法，请参见 《参考手册：命令》。

选择和创建口令

在添加用户作为 Adaptive Server 的登录名时，系统安全员将使用 create login 为每个用户分配一个口令。用户可使用 alter login 语句随时修改其口令。请参见第 74 页的 “更改口令”。

创建口令时：

- 不要使用诸如生日、街道名或其它任何与个人生活有关的单词或数字。
- 不要使用宠物或爱人的名字。
- 不要使用字典中出现的单词或反向拼写的单词。

最难猜的口令是那些组合了大小写字母和数字的口令。绝不要将口令泄露给他人或记录在别人能看到的地方。

口令必须满足以下条件：

- 长度至少为 6 个字符。 Sybase 建议使用 8 个字符或更长的口令。
- 包含任意可打印字母、数字或符号。
- 如果存在以下情况，则必须在 create login 中放在引号内：
  - 口令中包含 A - Z、a - z、0 - 9、_、#、有效的单字节或多字节字母字符、变音字母字符以外的任何字符
  - 以数字 0 - 9 开头

请参见第 24 页的 “口令复杂程度检查”。

设置和更改最大登录尝试次数

设置允许的最大登录尝试次数可以防止 “强制侵入”或以基于字典猜口令的方式尝试登录。系统安全员可以指定允许的最大连续登录尝试次数，尝试指定次数后，登录名或角色将被自动锁定。允许的失败登录尝试次数可以设置为适用于整个服务器或个别登录名和角色。个别设置优先于服务器范围的设置。

失败的登录次数存储在 master..syslogins 的 logincount 列中。成功的登录可以将失败登录次数重置为 0。

❖ **设置全服务器范围的 *maximum failed logins***

缺省情况下， maximum failed logins 处于禁用状态，不对口令应用此项检查。使用 sp_passwordpolicy 为登录名和角色设置全服务器范围的最大登录失败次数。

- 若要设置允许的失败登录次数，请输入：

```
sp_passwordpolicy 'set', 'maximum failed logins',
'number'
```

请参见 《参考手册：过程》中的 sp_passwordpolicy。

❖ **为特定登录名设置 *maximum failed logins***

- 若要在创建特定的登录名时为其设置最大登录尝试失败次数，请使用 create login。

此示例新建口令为 “Djdiek3”的登录名 “joe”，并将最大尝试登录失败次数设置为 3：

```
create login joe with password Djdiek3 max failed
attempts 3
```

请参见 《参考手册：命令》中的 create login。

❖ **为特定角色设置 *maximum failed logins***

- 若要在创建特定角色时设置 maximum failed logins，请使用 create role。

此示例创建口令为 “temp244” 的 “intern_role”，并将 “intern_role” 的 maximum failed logins 设置为 20：

```
create role intern_role with passwd "temp244", maximum
failed logins 20
```

请参见 《参考手册：命令》中的 create role。

❖ **更改特定登录名的 *maximum failed logins***

- 使用 alter login 设置或更改现有登录名的 maximum failed logins。

将登录名 “joe” 的 maximum failed logins 更改为 40：

```
alter login joe max failed attempts 40
```

❖ **更改特定角色的 *maximum failed logins***

- 使用 alter role 设置或更改现有角色的 maximum failed logins。

此示例将 “physician_role” 的允许 maximum failed logins 更改为 5：

```
alter role "all overrides" set maximum failed logins -1
```

此示例删除所有角色的 maximum failed logins 的替换值：

```
alter role physician_role set maximum failed logins 5
```

有关使用 maximum failed logins 的语法和规则的详细信息，请参见 《参考手册：命令》中的 alter role。

在丢失口令后登录

使用 dataserver -p*login_name* 参数可以在服务器启动时指定系统安全员或系统管理员的名称。如果无法恢复丢失的口令，则使用这种方法可以为这些帐户设置新口令。

使用 -p 参数启动时， Adaptive Server 会生成一个随机口令，接着显示并加密该口令，然后将其作为该帐户的新口令保存在 master..syslogins 中。

可以使用 dataserver -p 重新设置 sa_role 和 sso_role 的口令。如果丢失了这些角色中任一角色的口令，而这些角色需要口令才能变为活动状态，请使用 dataserver -p。

例如，如果使用以下命令启动服务器：

```
dataserver -psa_role
```

Adaptive Server 会显示以下消息：

```
New password for role 'sa_role' :qjcdyrbfkxgyc0
```

如果 sa_role 没有口令，且使用 -psa_role 启动，则 Adaptive Server 会在错误日志中显示一条错误消息。

Sybase 强烈建议您在重新启动服务器时更改登录名或角色的口令。

显示口令信息

本节论述如何显示登录名和角色的口令信息。

❖ **显示特定登录名的口令信息**

- 使用 sp_displaylogin 显示登录名的登录名和口令设置。

此示例显示有关绑定到登录配置文件的登录名 joe 的信息：

```
sp_displaylogin joe
Suid:3
Loginame:joe
Fullname:Joe Williams
Configured Authorization:
      sa_role (default ON)
      sso_role (default ON)
      oper_role (default ON)
Locked:NO
Date of Last Password Change:Sep 22 2008  3:50PM
Password expiration interval:0
Password expired:NO
Minimum password length:6
Maximum failed logins:1
Current failed login attempts:2
Authenticate with:ANY
登录配置文件:emp_lp
```

此示例显示有关未绑定到登录配置文件的登录名 joe 的信息：

```
sp_displaylogin joe
Suid:3
Loginame:joe
Fullname:
Default Database:master
Default Language:
Auto Login Script:
Configured Authorization:
Locked:NO
```

```
Date of Last Password Change:Sep 22 2008  3:50PM
Password expiration interval:0
Password expired:NO
Minimum password length:6
Maximum failed logins:1
Current failed login attempts:2
Authenticate with:ANY
Login Password Encryption:SHA-256
Last login date:Sep 18 2008 10:48PM
```

请参见 《参考手册：过程》中的 sp_displaylogin。

❖ **显示特定角色的口令信息**

- 使用 sp_displayroles 显示角色的登录名和口令设置。

此示例显示有关 physician_role 角色的信息：

```
sp_displayroles physician_role, "display_info"
Role name = physician_role
Locked:NO
Date of Last Password Change:Nov 24 1997  3:35PM
Password expiration interval = 5
Password expired:NO
Minimum password length = 4
Maximum failed logins = 10
Current failed logins = 3
```

请参见 《参考手册：过程》中的 sp_displayroles。

检查口令中是否至少包含一个数字

系统安全员可以指示服务器使用全服务器范围的配置参数 check password for digit，来检查口令中是否至少包含一个数字。如果设置了此参数，并不会影响现有的口令。缺省情况下，数字检查是关闭的。

此示例激活检查口令功能：

```
sp_configure "check password for digit", 1
```

这将停用检查口令功能：

```
sp_configure "check password for digit", 0
```

请参见 《参考手册：过程》中的 sp_configure。

设置和更改 *minimum password length*

可配置的口令允许自定义口令来适应需要，例如使用四位个人标识号 (PIN) 或口令为空值的匿名登录名。

注释　Adaptive Server 为 minimum password length 使用缺省值 6。Sybase 建议您对此参数使用 6 个或更多字符。

系统安全员可指定：

- 全局强制的 minimum password length
- 每个登录名或角色的 minimum password length

每个登录名或每个角色的值将替换全服务器范围的值。设置 minimum password length 只会影响设置该值后新创建的口令。

❖ **为特定登录名设置 *minimum password length***

- 若要在创建特定登录名时设置 minimum password length，请使用 create login。

此示例新建口令为 “Djdiek3”的登录名 “joe”，并将 “joe”的 minimum password length 设置为 8：

```
create login joe Djdiek3 with password @minpwdlen min
password length 8
```

请参见 《参考手册：命令》中的 create login。

❖ **为特定角色设置 *minimum password length***

- 若要在创建特定角色时设置 minimum password length，请使用 create role。

此示例创建一个口令为 “temp244”的新角色 “intern_role”，并将 “intern_role”的 minimum password length 设置为 0：

```
create role intern_role with passwd "temp244", min
passwd length 0
```

原口令为 7 个字符，但由于 minimum password length 设置为 0，因此该口令可更改为任意长度。

请参见 《参考手册：命令》中的 create role。

❖ **更改特定登录名的 *minimum password length***

- 使用 alter login 设置或更改现有登录名的 minimum password length。

此示例将登录名 “joe” 的 minimum password length 更改为 8 个字符。

```
alter login joe modify min password length 8
```

请参见 《参考手册：命令》中的 alter login。

❖ **更改特定角色的 *minimum password length***

- 使用 alter role 设置或更改现有角色的 minimum password length。

此示例将现有角色 “physician_role” 的最短长度设置为 5 个字符：

```
alter role physician_role set min passwd length 5
```

此示例替换所有角色的 minimum password length：

```
alter role "all overrides" set min passwd length -1
```

请参见 《参考手册：命令》中的 alter role。

❖ **删除特定登录名的 *minimum password length***

- 使用 alter login 删除现有登录名的 minimum password length。

此示例删除登录名 “joe” 的所有 minimum password length 限制：

```
alter login joe modify drop min password length
```

请参见 《参考手册：命令》中的 alter login。

口令复杂程度检查

可以在存储过程接口中使用这些支持口令复杂程度检查的选项；它们的值存储在 master.dbo.sysattributes 表中。

若要禁用某个单独选项，请输入：

sp_passwordpolicy 'clear', *option*

若要禁用所有口令策略选项，请输入：

sp_passwordpolicy 'clear'

登录口令复杂程度检查还扩展到角色口令。请参见第 157 页的 “对角色口令进行的登录口令策略检查”。

有关完整的 sp_passwordpolicy 语法，请参见 《参考手册：过程》。

不允许使用简单口令

disallow simple password 检查口令中是否包含了作为子字符串的登录名。可对其进行设置以便：

- 0 — （缺省值）禁用该选项，并允许简单口令。
- 1 — 启用该选项，并不允许使用简单口令。

要设置该选项，可输入：

```
sp_passwordpolicy 'set', 'disallow simple passwords',
      '1'
```

自定义口令复杂程度检查

Adaptive Server 允许您使用 sp_extrapwdchecks 和 sp_cleanpwdchecks 自定义配置口令检查规则。

这两个存储过程在 master 数据库中定义并位于该数据库中，分别在进行 Adaptive Server 口令复杂程度检查期间和删除登录名时自动调用。有关如何创建这两个自定义存储过程的示例，请参见第 30 页的 “启用自定义口令检查”。

指定口令中的字符

使用这些 sp_passwordpolicy 参数来指定口令中字符 （数字、大小写字符等）的最小数目：

- min digits in password — 口令中数字的最小数目。缺省为禁用。有效值包括：
  - 0 到 16 — 口令中至少必须包含的数字位数。
  - -1 — 口令中不能包含数字。
- min alpha in password — 口令中允许使用的字母字符的最小数目。该值不得小于大写字母最小数目与小写字母最小数目的和。缺省为禁用。有效值包括：
  - 0 到 16 — 口令中需要包含的特殊字符的最小数目。
  - -1 — 口令中不能包含特殊字符。
- min special char in password — 口令中特殊字符的最小数目。有效值包括：
  - 0 到 16 — 口令中需要包含的特殊字符的最小数目。
  - -1 — 口令中不能包含特殊字符。

- min upper char in password — 口令中大写字母的最小数目。缺省为禁用。有效值包括：
  - 0 到 16 — 口令中需要包含的大写字母的数目。
  - -1 — 口令中不能包含大写字母。
- min lower char in password — 口令中小写字母的最小数目。有效值包括：
  - 0 到 16 — 口令中需要包含的大写字母的数目。
  - -1 — 口令中不能包含大写字母。
- minimum password length — 口令的最小长度。可以将口令的最小长度设置在 0 - 30 之间。指定的值不得小于所有其它最小要求值的和。例如，如果已经进行了如下设置，则必须至少将 minimum password length 设置为 10：
  - minimum digits in password 设置为 3
  - minimum special characters in password 设置为 2
  - minimum uppercase characters in password 设置为 2
  - minimum lowercase characters in password 设置为 3
- password expiration — 口令到期之前可以使用多少天。指定的值将在全局范围内生效。缺省为禁用。有效值包括：
  - 0 — 口令永不过期。
  - 1 到 32767 — 口令到期之前可以使用的天数。
- password exp warn interval — 系统将在口令到期之前多少天显示口令有效期警告消息。这些消息将在每次成功登录时显示，直到更改了口令或口令到期。该值必须小于或等于口令有效期天数。缺省为禁用。

  有效值为 0 到 365。
- maximum failed logins — 指定失败登录的最大次数，在此次数之后登录名将被锁定。在全局范围内指定此值。缺省为禁用。有效值包括：
  - 0 — 无论登录失败多少次，始终不锁定登录名。
  - 1 到 32767 — 允许的失败登录次数，在此次数之后登录名将被锁定。

- expire login 用于在系统安全员创建或重置登录名时将登录名状态更改为已到期。这样，在首次登录时即需要为登录名更改口令。缺省为禁用。有效值包括：
  - 0 — 新登录名或重置的登录名将不到期。
  - 1 — 新登录名或重置的登录名到期；您必须在首次登录时重置口令。

请参见 《参考手册：过程》中的 sp_passwordpolicy。

口令复杂程度选项交叉检查

某些口令复杂程度选项相互影响：

- minimum password length 不得小于 min digits in password、 min alpha in password 和 min special characters in password 之和。
- min alpha in password 不得小于 min upper char in password 与 min lower char in password 的和。
- systemwide password expiration 必须大于 password exp warn interval。

为了进行上述交叉检查，如果 Adaptive Server 遇到的口令复杂程度选项值为 -1，则它会将该选项的值解释为 0。如果未设置某个选项，Adaptive Server 也会将该选项的值解释为 0。

Adaptive Server 将对每个不满足交叉检查的新口令复杂程度选项输出警告消息。不过，选项的设置仍然是成功的。

设置口令复杂程度检查

表 3-2：口令复杂程度检查

用于 Adaptive Server 鉴定的口令检查和策略	使用 sp_configure 指定的配置参数	使用 sp_passwordpolicy 指定的口令复杂程度选项	使用指定的每个登录名替换值 alter login
口令有效期	system-wide password expiration	system-wide password expiration	password expiration
口令中包含的数字	check password for digit	min digits in password	不适用
口令中包含的字母字符	不适用	min alpha in password	不适用
口令长度	minimum password length	minimum password length	min passwd length
失败登录的锁定	maximum failed logins	maximum failed logins	max failed attempts
不允许使用简单口令	不适用	disallow simple passwords	不适用
口令中包含的特殊字符	不适用	min special char in password	不适用
口令中包含的大写字母	不适用	min upper char in password	不适用
口令中包含的小写字母	不适用	min lower char in password	不适用

用于 Adaptive Server 鉴定的口令检查和策略	使用 sp_configure 指定的配置参数	使用 sp_passwordpolicy 指定的口令复杂程度选项	使用指定的每个登录名替换值 alter login
口令有效期警告间隔	不适用	password exp warn interval	不适用
首次登录时重置口令	不适用	expire login	不适用
自定义口令复杂程度检查	不适用	不适用	不适用

在以下级别设置口令复杂程度选项：

- 登录名级别，使用 create login 或 alter login。
- 全局级别，使用新的 sp_passwordpolicy 或 sp_configure。

由于设置的口令配置选项既可能是全局范围的，也可能是特定于每个登录名的，而且既可能使用旧参数，也可能使用新参数，因此将应用的口令选项的优先级顺序非常重要。

应用口令选项时，优先级顺序为：

1 每个登录名的现有参数

2 口令复杂程度选项

3 现有全局口令选项

示例

示例 1 创建新的登录名，并将 “johnd” 的 minimum password length 设置为 6：

```
create login johnd with password complex_password min
      password length '6'
```

登录名 “johnd” 的这些全局选项为登录名 “johnd” 创建两个最小口令长度要求，并设置有关口令中的数字的限制：

```
sp_configure 'minimum password length', '8'
sp_configure 'check password for digit', 'true'
sp_passwordpolicy 'set', 'min digits in password', '2'
```

如果之后尝试改变登录名 “johnd” 的口令：

```
alter login johnd with password complex_password modify
password 'abcd123'
```

Adaptive Server 将按如下顺序检查口令：

1　特定于每个登录名的现有选项检查：口令的最小长度必须大于 6。由于情况的确如此，因此通过检查。

2　新选项：口令中包含数字的最小数目必须大于 2。由于情况的确如此，因此通过检查。

3　现有全局选项：不检查此处指定的最小口令长度，因为已经对登录名 “johnd” 进行了特定于每个登录名的检查。

4　check password for digit 选项是冗余的，因为在打开数字的最小数目选项并设为 2 时，即已经对 check password for digit 选项进行了检查。

Adaptive Server 按指定顺序执行完这些检查之后，登录名 “johnd” 的新口令通过了这些检查，从而成功创更改了这个口令。

示例 2　如果为用户 “johnd” 输入了以下内容，则 Adaptive Server 首先会检查每个登录名的现有选项，确定最小口令长度设置为 6，而您尝试将口令改为仅使用 4 个字符：

```
alter login johnd with password complex_password modify
      password abcd
```

检查失败， Adaptive Server 输出一条错误消息。在一个口令复杂程度检查失败后，将不再对其它选项进行检查。

示例 3　使用以下口令配置选项创建新登录名，并将登录名 `johnd`的 minimum password length 设置为 4：

```
create login johnd with password complex_password min
      password length 4
```

这是每个登录名的现有选项。如果添加以下内容，则会创建口令中必须至少包含 1 个数字的全局要求：

```
sp_passwordpolicy 'set', 'min digits in password', '1'
```

如果之后尝试改变登录名 `johnd` 的口令，如下所示：

```
alter login johnd with password complex_password modify
      password abcde
```

Adaptive Server 将按如下顺序进行检查：

1　特定于每个登录名的现有选项检查：新口令的最小长度为 4。口令 “abcde” 的长度大于 4，因此通过检查。

2　新的全局要求检查：口令中所包含数字的最小数目被全局性地设为 1。该检查失败。

Adaptive Server 不更改口令，并输出一条错误消息。

若要改变口令，必须通过所有检查。

启用自定义口令检查

Adaptive Server 允许系统安全员编写启用自定义口令检查的用户定义存储过程。

例如，为实现口令历史记录检查，可以创建一个新的用户表来存储口令历史：

```
create table pwdhistory
(
      name varchar(30)not null,  – Login name.
      password varbinary(30)not null,  – old password.
      pwdate datetime not null,  – datetime changed.
      changedby varchar(30)not null  – Who changed.
)
go
```

指定新的口令时可以调用此用户定义存储过程 (sp_extrapwdchecks)，将口令以加密形式保存在 pwdhistory 表中：

```
create proc sp_extrapwdchecks
(
@caller_password varchar(30), – the current password of caller
@new_password    varchar(30), – the new password of the target acct
@loginame        varchar(30), – user to change password on
)
as

begin
declare @current_time    datetime,
        @encrypted_pwd   varbinary(30),
        @changedby       varchar(30),
        @cutoffdate      datetime


select @changedby = suser_name()

-- Change this line according to your installation.
-- This keeps history of 12 months only.
select @current_time = getdate(),
       @cutoffdate = dateadd(month,-12,getdate())
select @encrypted_pwd = hash(@new_password, 'sha1')

delete master..pwdhistory
    where name = @loginame
    and    pwdate < @cutoffdate

if not exists ( select 1 from master..pwdhistory
```

```
               where name = @loginame
                and   password = @encrypted_pwd )
begin
      insert master..pwdhistory
      select @loginame, hash(@new_password, 'sha1'),
             @current_time, @changedby
      return(0)
end
else
begin
      raiserror 22001  --user defined error message
end
end
```

使用 sp_addmessage 添加用户定义的消息 22001。raiserror 22001 指示自定义口令复杂程度检查错误。

可以使用以下用户定义的存储过程 (sp_cleanpwdchecks)，通过 sp_extrapwdchecks 来清除口令历史记录。

```
create proc sp_cleanpwdchecks
(
            @loginame      varchar(30)
                        -- user to change password on
)
as
begin

delete master..pwdhistory
where name = @loginame
end
      go
```

定义了上述两个过程并将它们安装在 master 数据库中之后，系统就会在口令复杂程度检查期间动态调用它们。

为口令设置登录名和角色有效期

系统管理员和系统安全员可以：

使用	目的
create login	在创建登录口令时指定其有效期。
alter login	更改登录口令的有效期。
create role	在创建角色口令时指定其有效期 （只有系统安全员才能发出 create role）。
alter role	在创建角色口令时更改其有效期 （只有系统安全员才能发出 alter role）。

以下规则适用于登录名和角色的口令有效期：

- 指派给个别登录帐户或角色的口令有效期将替换全局口令有效期的值。这使您可以为敏感的帐户或角色 （例如系统安全员口令）指定较短的有效期，而为不敏感的帐户 （例如匿名登录名）指定较为宽松的间隔。
- 口令到期的登录名或角色不能直接激活。
- 口令将在经过 password expiration interval 指定的天数后上次更改口令的时间到期。

有关命令和系统过程的语法和规则的详细信息，请参见相应的 《参考手册》。

12.x 版本以前的口令取消有效期限制

口令有效期不影响 Adaptive Server 12.x 之前版本中的角色。在 Adaptive Server 12.x 及更高版本中，任何现有用户定义角色的口令均取消口令有效期限制。

回避口令保护

在自动登录系统中，回避口令保护机制是必要的。可以创建不用口令即可访问其它角色的角色。

系统安全员可通过将口令保护的角色授予另一个角色来为某些用户回避口令机制，并将口令保护的角色授予一个或多个用户。激活此角色将会自动激活口令保护的角色，而不必提供口令。

例如：

Jane 是 ABC Inc. 的系统安全员，该公司使用自动登录系统。Jane 创建如下角色：

- financial_assistant

```
create role financial_assistant with passwd "L54K3j"
```

- accounts_officer

```
create role accounts_officer with passwd "9sF6ae"
```

- chief_financial_officer

```
create role chief_financial_officer
```

Jane 将 financial_assistant 和 accounts_officer 角色授予 chief_financial_officer 角色：

```
grant role financial_assistant, accounts_officer to
chief_financial_officer
```

然后， Jane 将 chief_financial_officer 角色授予 Bob：

```
grant role chief_financial_officer to bob
```

Bob 登录到 Adaptive Server 并激活 chief_financial_officer 角色：

```
set role chief_financial_officer on
```

financial_assistant 和 accounts_officer 角色将自动激活，无须 Bob 提供口令。现在，Bob 可以访问 financial_assistant 和 accounts_officer 角色下的所有内容，而无须输入这些角色的口令。

为新登录名创建口令有效期

使用 create login 为新登录名设置口令有效期。

此示例创建口令为 “Djdiek3” 的新登录名 “joe”，并将 “joe” 的口令有效期设置为 2 天：

```
create login joe with password Djdiek3 password
expiration 30
```

“joe” 的口令在创建登录帐户之日起 30 天后或自上次更改口令起 30 天后过期。

请参见 《参考手册：过程》中的 create login。

为新角色创建口令有效期

使用 create role 为新角色设置口令有效期。

此示例新建口令为 “temp244”的角色 intern_role，并将 intern_role 的口令有效期设置为 7 天：

```
create role intern_role with passwd "temp244", passwd expiration 7
```

intern_role 的口令将在您创建该角色之日起 7 天后或自上次更改口令起 2 天后过期。

请参见 《参考手册：命令》中的 create role。

为口令添加的创建日期

口令使用创建日期标记，该日期等于给定服务器的升级日期。登录口令的创建日期存储在 syslogins 的 pwdate 列中。角色口令的创建日期存储在 syssrvroles 的 pwdate 列中。

更改或删除登录名或角色的口令有效期

使用 alter login 更改现有登录名的口令有效期，为没有口令有效期的登录名增加口令有效期，或删除口令有效期。 alter login 只影响登录口令，并不影响角色口令。

此示例将登录名 “joe”的口令有效期更改为 5 天：

```
alter login joe modify password expiration 30
```

口令将在您实施口令到期之日起 30 天后到期。

请参见 《参考手册：命令》中的 alter login。

保护网络上的登录口令

Adaptive Server 允许使用非对称加密，以采用 RSA 公用密钥加密算法将口令从客户端安全地传输到服务器。Adaptive Server 生成非对称密钥对，并将公用密钥发送给使用登录协议的客户端。例如，在将用户登录口令发送到服务器之前，客户端将使用公用密钥对该口令进行加密。服务器则使用私有密钥将口令解密，以开始对客户端连接进行鉴定。

您可以将 Adaptive Server 配置为要求客户端使用登录协议。将 Adaptive Server 配置参数 net password encryption reqd 设置为要求所有基于用户名和口令的鉴定请求使用 RSA 非对称加密。请参见 《系统管理指南第一卷》的第 5 章 “设置配置参数”中的 “net password encryption required”。

生成非对称密钥对

在以下情况下， Adaptive Server 生成新的密钥对：

- 服务器每次启动时，
- 使用 Adaptive Server 管家机制每隔 24 小时自动生成一次，
- 具有 sso_role 的管理员请求重新生成密钥对时。

密钥对保存在内存中。重新生成密钥对时，错误日志和审计追踪中会记录一条消息。

要在需要时生成密钥对，请使用：

```
sp_passwordpolicy "regenerate keypair"
```

注释　根据系统负载，此命令执行的时间与密钥对实际生成的时间之间可能有一段延迟。这是因为管家任务以低优先级运行，且可能由于更高优先级的任务而被延迟。

要在具体时间生成密钥对，请使用：

```
sp_passwordpolicy "regenerate keypair", datetime
```

其中 *datetime* 是要重新生成密钥对的日期和时间。

例如，日期时间字符串 “Jan 16, 2007 11:00PM” 会在该指定时间生成密钥对。日期字符串也可以只是一天中的某个时间，例如 “4:07AM”。如果仅指定了一天中的某个时间，重新生成密钥对会安排在 24 小时之后的该时间点。

sp_passwordpolicy 能让您配置密钥对重新生成的频率，以及在密钥对重新生成失败时 Adaptive Server 应该做什么：

- ‘keypair regeneration period’, { ([*keypair regeneration frequency*], datetime of first generation) | (keypair regeneration frequency, [*datetime of first generation*]) }
- “keypair error retry [wait | count]”, “*value*”

请参见 《参考手册：系统过程》中的 sp_passwordpolicy。

服务器选项 “net password encryption”

建立远程过程调用 (RPC) 时， Adaptive Server 也充当客户端。

当连接到远程服务器时， Adaptive Server 使用 net password encryption 选项来确定是否使用口令加密。

当此服务器选项设置为 true 时，Adaptive Server 使用 RSA 或 Sybase 专有算法。用于启用 net password encryption 的命令为：

```
sp_serveroption server, "net password encryption",
    "true"
```

此设置存储在 master..sysservers 中，可以使用 sp_helpserver 存储过程显示服务器选项的值。

对于使用 sp_addserver 添加的任何新服务器， net password encryption 的缺省值为 true。在升级过程中，对于含有 ASEnterprise 类值的 sysservers 条目，Adaptive Server 将 net password encryption 设置为 true。不对其它服务器类进行修改。这样可以提高两个通信的 Adaptive Server 之间的口令安全性。

注释 如果您在与服务器建立连接时遇到问题，管理员可以选择将 net password encryption 重置为 false。但是，如果该选项设置为 false，口令就会在网络中以明文传输。

向后兼容性

- Sybase 建议您使用 RSA 算法保护网络上的口令。
- 若要使用 RSA 算法，必须有 Adaptive Server 15.0.2 版和新的 Connectivity SDK 客户端 （15.0 ESD#7 版及更高版本）。 Sybase 提供 net password encryption reqd 配置参数和 net password encryption 服务器选项，以允许使用等效于 15.0.2 之前的版本设置，并保持与旧客户端和服务器的向后兼容性。
- 不支持 RSA 算法的旧客户端可将此属性设置为使用 Sybase 专有算法进行口令加密，此算法已在 12.0 版中提供。然后， Adaptive Server 会使用 Sybase 专有算法。
- 支持 RSA 和 Sybase 专有算法的新客户端可以为这两种算法设置属性。与此类客户端通信时， Adaptive Server 15.0.2 使用 RSA 算法。15.0.2 之前版本的 Adaptive Server 使用 Sybase 专有算法。

保护磁盘上和内存中存储的登录口令

Adaptive Server 用于鉴定客户端连接的登录口令以 SHA-256 散列摘要的形式安全地存储在磁盘上。 SHA-256 算法是一种单向加密算法，它生成的摘要不能被解密，确保了存储在磁盘上的安全性。为了鉴定用户的连接，会对客户端发送的口令应用 SHA-256 算法，然后将结果与磁盘上存储的值进行比较。

为了防止对磁盘上存储的登录口令进行基于字典的攻击，在使用 SHA-256 算法之前，将一个 salt 与口令混合。 salt 与 SHA-256 散列存储在一起，且在登录鉴定过程中使用。

Sybase 建议：只要您确保将不会降级到早期版本，即仅使用 SHA-256。在进行决策时请考虑权衡；如果需要降级到 15.0.2 之前的版本，则该算法需要管理员的干预才能将用户登录口令解锁。

口令的字符集注意事项

被加密的口令和其它敏感数据必须确定明文字符集，才能准确地解释鉴定过程中解密的结果或者散列值的比较结果。

例如，客户端使用 isql 连接到 Adaptive Server 并建立新的口令。无论客户端使用的是什么字符集，字符通常都会转换为服务器的缺省字符集，以用于在 Adaptive Server 中进行处理。假设 Adaptive Server 的缺省字符集为 “iso_1”，考虑命令：

```
alter login loginName with password oldPasswd modify password
newPasswd
```

该口令参数为 varchar，以带引号的字符串表示，在加密前以 “iso_1” 编码存储。如果 Adaptive Server 的缺省字符集以后进行更改，则加密的口令仍为以原始缺省字符集编码的加密字符串。由于不匹配的字符映射，这可能导致鉴定失败。尽管缺省字符集很少更改，但在平台间进行迁移时变得尤为重要。

Adaptive Server 在加密前将明文口令转换为规范形式，以便可跨平台、芯片体系结构和字符集使用口令。

为 syslogins 中的存储使用规范形式：

1　将明文口令字符串转换为 UTF-16。

2　将 UTF-16 字符串转换为网络字节顺序。

3　将具有随机字节的小缓冲区 (salt) 附加到口令。

4　应用 SHA-256 散列算法。

5　在 password 列中存储摘要、 salt 和版本。

在鉴定时：

1　将明文口令字符串转换为 UTF-16。

2　将 UTF-16 字符串转换为网络字节顺序。

3　将 syslogins 中的 password 列中的 salt 附加到口令。

4　应用散列算法。

5　将结果与 syslogins 中的口令列进行比较，如果匹配，则鉴定成功。

升级和降级行为

本节包含有关在各版本之间升级和降级 Adaptive Server 的信息。

注释 如果您要从 Adaptive Server 15.0 版升级到 15.7 版，请查看本节内容。

登录口令降级

为了简化在从 15.0.2 之前版本迁移时向新的磁盘上加密算法的过渡，Adaptive Server 包括了口令策略 allow password downgrade。从 15.0.2 之前版本升级后，该策略的值为 1，表示口令既会以早期版本使用的 Sybase 专有算法存储，也会以 Adaptive Server 15.0.2 和更高版本中使用的 SHA-256 算法存储。

只要口令以新旧两种形式存储，则无需重置用户口令，就可以将 Adaptive Server 降级到 Adaptive Server 15.0。如果将策略 allow password downgrade 设置为 0，则口令只会以新的 SHA-256 形式存储，此形式与旧版本不兼容。当降级到先前版本时，只有以 SHA-256 存储的口令会被重新设置为随机口令，并会以与旧版本兼容的旧形式存储。

要不再允许口令降级，请执行：

```
sp_passwordpolicy 'set', 'allow password downgrade',
'0'
```

在执行此命令前，请使用 sp_displaylogin 检查登录帐户，以确定登录帐户是否已使用，以及口令是否以 SHA-256 编码存储。否则，将会自动锁定该帐户，并用生成的口令重置帐户。若要再次使用该帐户，您必须对该帐户进行解锁，并为用户提供新生成的口令。

您可能需要保存此命令的输出，因为其中可能包含有关被锁定的登录帐户及这些帐户的生成口令的信息。

口令降级阶段结束时：

- 当口令降级阶段结束时，datetime 会记录在 master.dbo.sysattributes 中。
- syslogins 中的每个 password 列的值都会被重写，以仅使用磁盘上结构的新口令。
- 如果登录名尚未过渡到新算法，则其口令会被重置为服务器生成的、SHA-256 格式的新口令，并且该登录名将被锁定。生成的口令仅显示给执行上述 sp_passwordpolicy 过程的管理员。锁定原因设置为 3（“未过渡到 SHA-256 的登录名或角色”）。

sp_passwordpolicy 过程完成后：

- 登录鉴定仅使用 SHA-256。
- 仅为 password 列使用磁盘上结构的新口令。
- 尝试使用锁定的登录名不会通过鉴定。若要使用锁定的登录名，您必须使用 sp_locklogin 解锁该登录名，且用户必须使用 sp_passwordpolicy 生成的口令。或者，您可能更愿意为锁定的登录帐户指定新的口令，而不是使用生成的口令。

示例 1

本示例准备了一个仅使用 SHA-256 的升级服务器。使用 sp_displaylogin 检查登录帐户以确定该帐户使用的是哪种加密。

```
1> sp_displaylogin login993
2> go
Suid:70
Loginame:login933
Fullname:
Default Database:master
Default Language:
Auto Login Script:
Configured Authorization:
Locked:NO
Date of Last Password Change:Apr 20 2007 2:55PM
Password expiration interval:0
Password expired:NO
Minimum password length:0
Maximum failed logins:3
Current failed login attempts:
Authenticate with:ANY
Login Password Encryption:SYB-PROP
Last login date:
(return status = 0)
```

Login Password Encryption: SYB-PROP 行中的 SYB-PROP 值指示为此帐户仅使用 Sybase 专有加密。此登录名在升级到 Adaptive Server 15.0.2 版和更高版本之前尚未使用过，并且，如果执行了 sp_passwordpolicy 'set', 'allow password downgrade', '0'，此登录名会被锁定，且其口令将被重置。

升级到 Adaptive Server 15.0.2 后首次登录到帐户之后，此行会发生更改，显示同时使用了新旧两种加密：

```
Login Password Encryption:SYB-PROP,SHA-256
```

这是所有活动的登录帐户的预期状态，因此执行 sp_passwordpolicy 'set', 'allow password downgrade', '0' 不会锁定帐户和重置帐户的口令。

执行 sp_passwordpolicy 'set', 'allow password downgrade', '0' 之后，将只会使用 SHA-256 加密，并且您将看到：

```
Login Password Encryption:SHA-256
```

显示此值的登录帐户现在使用更强大的磁盘上加密算法。

当所有口令都更改为使用新的算法时，重新执行 sp_passwordpolicy 将不会显示重置或锁定的帐户：

```
1> sp_passwordpolicy 'set', 'allow password downgrade', '0'
2> go

Old password encryption algorithm usage eliminated from 0 login accounts,
changes are committed.
(return status = 0)
```

示例 2

此示例中， 1000 个登录帐户中有 990 个都已过渡到 SHA-256 算法，但有 10 个帐户仍在使用 SYB-PROP 算法：

```
1> sp_passwordpolicy 'set', 'allow password downgrade', '0'
2> go

Old password encryption algorithm found for login name login1000, suid 3,
ver1 =5, ver2 = 0, resetting password to EcJxKmMvOrDsC4
Old password encryption algorithm found for login name login999, suid 4,
ver1 =5, ver2 = 0, resetting password to MdZcUaFpXkFtM1
Old password encryption algorithm found for login name login998, suid 5,
ver1 =5, ver2 = 0, resetting password to ZePiZdSeMqBdE6
Old password encryption algorithm found for login name login997, suid 6,
ver1 =5, ver2 = 0, resetting password to IfWpXvGlBgDgW7
Old password encryption algorithm found for login name login996, suid 7,
ver1 =5, ver2 = 0, resetting password to JhDjYnGcXwObI8
Old password encryption algorithm found for login name login995, suid 8,
ver1 =5, ver2 = 0, resetting password to QaXlRuJlCrFaE6
Old password encryption algorithm found for login name login994, suid 9,
ver1 =5, ver2 = 0, resetting password to HlHcZdRrYcKyB2
Old password encryption algorithm found for login name login993, suid 10,
ver1 =5, ver2 = 0, resetting password to UvMrXoVqKmZvU6
Old password encryption algorithm found for login name login992, suid 11,
ver1 =5, ver2 = 0, resetting password to IxIwZqHxEePbX5
Old password encryption algorithm found for login name login991, suid 12,
ver1 =5, ver2 = 0, resetting password to HxYrPyQbLzPmJ3
Old password encryption algorithm usage eliminated from 10 login accounts,
changes are committed.
```

```
(return status = 1)
```

注释　登录名、 suid 和生成的口令将向执行过程的管理员显示。此命令的输出表明所有 10 个未过渡的帐户都被重新设置 （并锁定）。

升级的 *master* 数据库中的行为更改

当您升级 master 数据库时， Adaptive Server 将使用 password 列中 Adaptive Server 早期或升级版本中的算法保留 syslogins 目录中的加密口令。

用户可以调用 sp_displaylogin 来确定某次登录使用哪种 “Login password encryption”。

在升级后进行首次登录鉴定时：

- 用户使用 password 列的内容和旧算法进行鉴定。
- Adaptive Server 会先后使用旧的加密算法和新的加密算法更新 password 列。

在升级后进行后续登录鉴定时，在 “allow password downgrade”设置为 0 之前，用户使用新的算法鉴定。

新的 *master* 数据库中的行为更改

在新的 Adaptive Server master 数据库中，或者在 allow password downgrade 设置为 0 之后的升级 master 数据库中，服务器仅使用 password 列中的新算法保留 syslogins 中的加密口令。只有 SHA-256 算法才会鉴定连接请求和在磁盘上存储口令。

发出 sp_passwordpolicy，以确定服务器是否已升级 （例如，从 15.0 版升级到 15.0.2 版），并使用升级前或升级后的服务器中的算法保留口令，或者，确定是否新安装了服务器，并且服务器是否包括使用 （15.0.2 版中的）最新算法的 master 数据库：

```
sp_passwordpolicy 'list', 'allow password downgrade'
```

在升级并随之降级后保留口令加密

如果升级到 Adaptive Server 15.0.2 或更高版本，然后降级到某个早期版本，请使用 sp_downgrade 以保留并使用 15.0.2 及更高版本服务器中的口令加密功能。缺省情况下，Adaptive Server 允许您在升级之后降级口令，直到结束口令降级阶段为止。

注释 如果运行 sp_downgrade，关闭服务器，然后重新启动您从其降级的 Adaptive Server 的同一版本，则会删除 sp_downgrade 所做的更改。必须重新运行 sp_downgrade 以重复更改。有关运行 sp_downgrade 的信息，请参见 《安装指南》。

在升级之前增加空间

Adaptive Server 要求 master 数据库和事务日志中有额外的空间。使用 alter database 为 master 数据库和事务日志增加额外的空间。

加密算法和口令策略：

- 将 syslogins 所需的空间增加大约 30%。
- 将每个登录帐户的最大行长度增加 135 个字节。
- 在 Adaptive Server 15.0.1 版和 15.0.2 版间，将每页行数的比例从每 2K 页 16 行降低到每 2K 页 12 行。在降级过程中有一段时间 allow password downgrade 为 1 （此时同时使用了新旧两种口令加密算法）；该比率会进一步降低到每 2K 页大约 10 行。

  例如，如果 Adaptive Server 15.0.1 有 1,000 个登录帐户，数据适合 59 页，则同样数目的登录帐户在 Adaptive Server 15.0.2 新的 master 数据库上可能另外需要大约 19 页；或者，如果从 15.0.1 升级 （allow password downgrade 设置为 1），则另外需要 33 页。

对于更新的 password 列，事务日志需要额外的空间。用户第一次登录时，Adaptive Server 需要每 1,000 个登录大约 829 个 2K 页，而对于用户在升级和降级过程中进行的口令更改，则需要每 1,000 个登录大约 343 页。若要确保有足够的日志空间，在开始口令升级或降低之前，以及在用户第一次登录到 Adaptive Server 15.0.2 版及更高版本时，请验证每个登录有大约一个 2K 页的可用日志空间。

降级

Adaptive Server 支持从 15.0.2 或更高版本降级到 15.0 或 15.0.1 版。如果要降级到更低版本的 Adaptive Server，则可能需要执行额外的操作。

如果 allow password downgrade 为 0 或 NULL，或者，如果口令已仅使用 SHA-256 算法存储在 syslogins 中，请对登录帐户使用 sp_displaylogin 以确定使用了哪种算法，或使用 sp_downgrade "prepare" 以确定重置了哪些帐户。

prepare 选项报告服务器是否已准备好降级。如果 prepare 选项失败，它将报告必须修复的错误。如果在修复错误之前在服务器上执行降级，则降级将失败。对于登录口令， prepare 将报告在降级过程中重置了哪些口令。

运行 sp_downgrade "prepare"，验证您是否应运行 sp_downgrade：

```
sp_downgrade 'prepare','15.0.1',1

Checking databases for downgrade readiness.

There are no errors which involve encrypted columns.

Allow password downgrade is set to 0. Login passwords
may be reset, if old encryption version of password is
not present.

Warning:New password encryption algorithm found for
login name user103, suid 103.

Password will be reset during the downgrade phase.

sp_downgrade 'prepare' completed.
 (return status = 0)

drop login probe
```

如果该登录名在数据库中有用户条目，请从 master 数据库中删除用户，然后删除该登录名：

```
use master
sp_dropuser 'probe'
```

对降级的服务器运行 installmaster 时，会重新创建 probe 登录名。

执行 sp_downgrade 之前，Sybase 建议删除 syslogins 和 syssrvroles 的统计信息。这样做可避免在降级过程记录 sysstatistics 中的无效列信息，例如口令列的长度。

若要删除 syslogins 和 syssrvroles 的统计信息，请输入：

```
delete statistics master..syslogins
delete statistics master..syssrvroles
```

在此示例中，执行 sp_downgrade 将会锁定并重置 user103 的登录口令。Adaptive Server 生成的随机口令仅对执行 sp_downgrade 的客户端显示。管理员可以将此输出重定向到文件，以便保留这些口令；管理员也可以在降级完成并且服务器重新启动后手动重置这些口令。

```
sp_downgrade 'downgrade','15.0.1',1

Checking databases for downgrade readiness.
There are no errors which involve encrypted columns.

Allow password downgrade is set to 0. Login passwords may be reset, if old
encryption version of password is not present.
Warning:New password encryption algorithm found for login name user103, suid
103 .
Password is reset during the downgrade phase.

Executing downgrade step 1 [sp_passwordpolicy 'downgrade'] for :
- Database:master (dbid:1)

New password encryption algorithm found for login name user103, suid 103.
Resetting password to 'ZdSuFpNkBxAbW9'.

Total number of passwords reset during downgrade = 1

[ ... output from other downgrade steps ..]
(return status = 0)
```

错误日志中将出现其他消息，标识在 sp_downgrade 过程中进行的步骤：

```
00:00000:00006:2007/05/21 05:34:07.81 server  Preparing ASE downgrade from 1502 to 1501.
00:00000:00006:2007/05/21 05:35:59.09 server  Preparing ASE downgrade from 1502 to 1501.
00:00000:00006:2007/05/21 05:35:59.19 server  Starting downgrading ASE.
00:00000:00006:2007/05/21 05:35:59.20 server  Downgrade :Downgrading login passwords.
00:00000:00006:2007/05/21 05:35:59.22 server  Downgrade :Starting password downgrade.
00:00000:00006:2007/05/21 05:35:59.23 server  Downgrade :Removed sysattributes rows.
00:00000:00006:2007/05/21 05:35:59.23 server  Downgrade :Updated 1 passwords.
00:00000:00006:2007/05/21 05:35:59.24 server  Downgrade :Removed columns in syslogins -
lastlogindate, crdate, locksuid, lockreason, lockdate are removed.
00:00000:00006:2007/05/21 05:35:59.26 server  Downgrade :Truncated password lengths.
00:00000:00006:2007/05/21 05:35:59.28 server  Downgrade :Successfully completed password
downgrade.
00:00000:00006:2007/05/21 05:35:59.28 server  Downgrade :Marking stored procedures to
be recreated from text.
00:00000:00006:2007/05/21 05:36:03.69 server  Downgrade :Dropping Sysoptions system
table.
```

```
00:00000:00006:2007/05/21 05:36:03.81 server  Downgrade :Setting master database minor
upgrade version.
00:00000:00006:2007/05/21 05:36:03.83 server  Downgrade :Setting user databases minor
upgrade version.
00:00000:00006:2007/05/21 05:36:03.90 server  ASE downgrade completed.
```

sp_downgrade 更改目录并修改口令数据。服务器必须为单用户模式，才能成功执行 sp_downgrade。若要以单用户模式启动服务器并仅允许系统管理员登录，请使用 -m 命令行选项来启动服务器。

运行 sp_downgrade 之后，请关闭 15.0.2 服务器，以避免新的登录或者其它可能修改数据或系统目录的操作。如果在运行 sp_downgrade 之后重新启动 Adaptive Server 15.0.2 版，则早期版本将关闭，并且您将再次升级到 15.0.2 版或更高级别。

当 *allow password downgrade* 设置为 0 时，口令会到期。

在口令降级阶段结束时使 syslogins 中的口令到期。

要将登录口令配置为到期，请使用：

```
sp_passwordpolicy "expire login passwords"[, "[loginame | wildcard]"
```

若要将角色口令配置为到期，请使用：

```
sp_passwordpolicy "expire role passwords"[, "[rolename | wildcard]"
```

要将旧登录名口令配置为到期，请使用：

```
sp_passwordpolicy "expire stale login passwords", "datetime"
```

要将旧角色口令配置为到期，请使用：

```
sp_passwordpolicy "expire stale role passwords", "datetime"
```

当您执行命令时，自 sp_passwordpolicy "expire stale login passwords," 的 *datetime* 参数中所设置日期以来未更改的口令将到期。当口令降级阶段结束时，将自动要求用户更改其口令。

也可以锁定旧登录名或角色；但是，这将要求您手动重置口令，以使合法用户可以再次访问其登录帐户。

显示 *allow password downgrade* 的当前值

若要获取 allow password downgrade 的当前值，请输入：

```
sp_passwordpolicy 'list', 'allow password downgrade'
```

结果集包括当前值及说明其意义的消息。

如果已升级 master 数据库，并且要采用新旧两种编码保留口令，则结果为：

```
sp_passwordpolicy 'list', 'allow password downgrade'
go

value    message
-------- -----------------------------------------------------
       1 Password downgrade is allowed.
(1 row affected)
```

对于仅使用新口令加密的升级 master 数据库，此结果为：

```
sp_passwordpolicy 'list', 'allow password downgrade'
go

value    message
-------- -----------------------------------------------------
       0 Last Password downgrade was allowed on <datetime>.
(1 row affected)
```

对于 Adaptive Server 15.0.2 上的仅使用新口令加密的新 master 数据库，此结果为：

```
sp_passwordpolicy 'list', 'allow password downgrade'
go

value    message
-------- -----------------------------------------------------
   NULL New master database.
(1 row affected)
```

在高可用性环境中使用口令

口令安全性影响高可用性的配置和 syslogins 中的口令在主服务器和协同服务器之间的行为。

高可用性配置

在您针对高可用性配置主服务器和协同服务器之前，这些服务器必须具有相同的 allow password downgrade 值。allow password downgrade 的 quorum 属性检查 allow password downgrade 的值在主服务器和辅助服务器上是否相同。

如果主服务器上的 allow password downgrade 为 1，在辅助服务器上为 0，则 sp_companion 的输出为：

```
1> sp_companion "primary_server",configure
2> go

Step:Access verified from Server:'secondary_server' to Server:'primary_server'.
Step:Access verified from Server:'primary_server' to Server:'secondary_server'.
Msg 18836, Level 16, State 1:
Server 'secondary_server', Procedure 'sp_companion', Line 392:
Configuration operation 'configure' can not proceed due to Quorum Advisory Check
failure.Please run 'do_advisory' command to find the incompatible attribute
and fix it.

Attribute Name          Attrib Type      Local Value   Remote Value   Advisory
--------------          -----------      -----------   ------------   --------
allow password downg    allow password             0              1          2

(1 row affected)
(return status = 1)
```

advisory 列中的值 2 表示：除非两个协同服务器上的值相同，否则用户不能进行集群操作。

sp_companion do_advisory 还会列出两个服务器上的 allow password downgrade 值的差异。

在主服务器和辅助服务器上单独运行 sp_passwordpolicy 'allow password downgrade' 以同步值，并确保服务器的状态相同。

升级后更新的口令

在针对高可用性升级和配置后第一次连接到主服务器时，在具有相同磁盘上加密格式的主服务器和协同服务器上，用户登录口令将同步。这样可避免在 allow password downgrade 阶段结束并且口令降级到 Adaptive Server 的早期版本时出现口令重置或锁定情况。无需被 sp_passwordpolicy 或 sp_downgrade 重置或锁定，就可以继续使用登录口令。

成功设置高可用性环境后，请分别在主服务器和协同服务器上结束 allow password downgrade 阶段。同样，降级到 Adaptive Server 的早期版本，分别在主服务器和协同服务器上执行 sp_downgrade。

建立口令和登录策略

Adaptive Server 包括几个控件，可用来设置登录、角色和口令的策略以进行内部鉴定。

在 Adaptive Server 中，系统安全员可以：

- 指定在某个登录名或角色被自动锁定前，为该登录名或角色键入无效口令的最大允许次数
- 在丢失口令后登录
- 手动记录和解锁登录名和角色
- 显示登录口令信息
- 指定服务器范围或特定登录或角色所需的 minimum password length。
- 检查登录名的口令复杂程度
- 启用登录名的自定义口令检查
- 设置口令有效期
- 考虑登录口令字符集
- 锁定不活动的登录帐户
- 在高可用性环境中使用口令

登录失败

Adaptive Server 必须成功对用户进行鉴定，该用户才能够访问 Adaptive Server 中的数据。如果鉴定尝试失败， Adaptive Server 将返回以下消息且网络连接将终止：

```
isql -U bob -P badpass
Msg 4002, Level 14, State 1:
Server 'ACCOUNTING'
Login failed.
CT-LIBRARY error:
ct_connect():protocol specific layer:external error:
The attempt to connect to the server failed
```

此消息是常规登录失败消息，它不会通知进行连接的用户失败是由错误的用户名还是由错误的口令导致。

尽管客户端显示登录失败常规消息的目的是为了避免向恶意用户提供信息，但系统管理员可能会发现，失败的原因对于检测入侵尝试和诊断用户鉴定问题十分重要。

Adaptive Server 在 sysaudits.extrainfo 列的 `Other Information` 部分的 `Errornumber.Severity.State` 中提供了登录失败原因。登录失败审计具有事件编号 45 和 eventmod 2。

将 sp_audit login 参数设置为 on 或 fail 以启用登录失败审计：

```
sp_audit "login", "all", "all", "fail"
sp_audit "login", "all", "all", "on"
```

请参见 “审计登录失败”。

锁定 Adaptive Server 登录帐户和角色

若要防止某个用户登录到 Adaptive Server，可以锁定或删除其 Adaptive Server 登录帐户。锁定登录帐户可以保留 suid 以使其无法重新使用。

执行 sp_locklogin 以锁定登录帐户

当锁定登录帐户时，会在 login_locked 审计选项下面生成具有审计事件 AUD_EVT_LOGIN_LOCKED (112) 的审计记录，因为登录尝试次数达到了所配置的 maximum failed login 值。

警告！ Adaptive Server 可能在创建下一个登录帐户时重新使用已删除的登录帐户的服务器用户 ID (suid)。这种情况只有在所删除的登录帐户持有 syslogins 中最大的 suid 的情况下才会出现；不过，如果不对 drop login 的执行情况进行审计，则会造成无法分清责任。此外，带有重新使用的 suid 的用户有可能可以访问授权给旧 suid 的数据库对象。

不能在以下情况下删除登录：

- 用户位于任何数据库中。
- 登录是最后一个保留下来的用户，该用户拥有系统安全员或系统管理员角色。

系统安全员可以使用 sp_locklogin 或 drop login 锁定或删除登录名。如果为复制记录了系统过程，则系统安全员在发出命令时必须位于 master 数据库中。

锁定和解锁登录名

以下情况中可以锁定登录名：

- 口令到期，或
- 达到最大登录尝试失败次数，或
- 系统安全员手动将其将锁定。

❖ **锁定和解锁登录名**

- 系统安全员可使用 sp_locklogin 手动锁定或解锁登录名。例如：

```
sp_locklogin "joe" , "lock"
sp_locklogin "joe" , "unlock"
```

有关登录锁定状态的信息存储在 syslogins 的 status 列中。

请参见 《参考手册：过程》中的 sp_locklogin。

锁定和解锁登录帐户

使用 sp_locklogin 锁定和解锁帐户或显示锁定帐户的列表。您必须是系统安全员才能使用 sp_locklogin。

语法为：

```
sp_locklogin  [ {login_name }, { "lock" | "unlock" } ]
```

其中：

- *login_name* 是要锁定或解锁的帐户的名称。登录名必须是一个已存在的有效帐户。
- all 表示锁定或解锁 Adaptive Server 上的所有登录帐户，具有 sa_role 的帐户除外。
- lock | unlock 指定是否要锁定或解锁帐户。

若要显示所有锁定登录的列表，请使用不带参数的 sp_locklogin。

可以锁定当前登录的帐户，但用户在注销之前不会受帐户锁定的影响。可以锁定数据库所有者的帐户，并且锁定的帐户也可以拥有数据库中的对象。此外，可以使用 sp_changedbowner，将某个锁定帐户指定为数据库所有者。

Adaptive Server 确保至少总是有一个未锁定的系统安全员帐户和一个未锁定的系统管理员帐户。

使用 *syslogins* 跟踪帐户是否已锁定

syslogins 包括 lastlogindate、 crdate、 locksuid、 lockreason 和 lockdate 列，以支持上次登录和锁定不活动的帐户，从而使帐户所有者或管理员能够了解帐户是否已锁定、帐户的锁定时间、锁定者以及锁定原因。

创建登录时， crdate 列设置为当前时间。

如果 enable last login updates 口令策略选项设置为 1，则 lastlogindate 列将设置为登录的 datetime，并且该列以前的值存储在登录会话的进程状态结构中。每次登录 Adaptive Server 时都会更新 syslogins 和进程状态结构。对于新的 master 数据库或升级的数据库， enable last login updates 的缺省值为 1。若要禁用此选项，请使用管理员权限执行此过程：

```
sp_passwordpolicy 'set', 'enable last login updates',
'0'
```

@@*lastlogindate* 特定于每次用户登录会话，该会话可以使用此变量来确定帐户上次登录的日期和时间。如果帐户之前未使用过，或 enable last login updates 为 0，则 @@*lastlogindate* 的值为 NULL。

事务日志不会记录对 syslogins..lastlogindate 的更新。

具有 sso_role 的管理员可以使用以下命令锁定在给定天数内处于不活动状态的登录帐户：

```
sp_locklogin 'lock', [@except], 'number of inactive days'
```

如果 enable last login updates 设置为 0，或 lastlogindate 列的值为 NULL，则此命令无效。 *number of inactive days* 的值范围为 1 - 32767 （天）。

lockreason 列指定锁定登录的原因。lockdate 列的值设置为当前 datetime。

如果帐户被解锁，列 lockreason、 lockdate 和 locksuid 都会被重新设置为 NULL。

lockdate、locksuid 和 lockreason 列由 Adaptive Server 内部设置。表 3-3 提供了 lockreason 的值和说明，以及 locksuid 的值。

表 3-3：原因和 locksuid 的值

lockreason 的值	locksuid 的值	帐户 lockreason 的说明
NULL	NULL	帐户未被锁定。
0	suid of caller of sp_locklogin	locksuid 通过执行 sp_locklogin 而手动锁定的帐户。
1	suid of caller of sp_locklogin	因帐户不活动而锁定的帐户， locksuid 已手动执行 sp_locklogin 'all', 'lock', 'ndays'。
2	suid of attempted login	因登录尝试失败次数达到了最大登录失败次数而由 Adaptive Server 锁定的帐户。
3	suid of caller of sp_passwordpolicy set, "allow password downgrade", 0	因口令降级阶段已结束、登录名或角色尚未过渡到 SHA-256 而由 locksuid 锁定的帐户。
4	NULL	帐户由于不活动而被锁定。

锁定和解锁角色

记帐信息（如何时锁定角色、为何锁定，以及由谁锁定）存储在 syssrvroles 中，可以用于角色锁定记帐。

锁定角色的原因可以有多种：

- 输入错误的角色口令达到指定的次数。“max failed_logins”选项可以在创建或改变角色时与角色相关联。它指定在角色激活尝试失败多少次之后锁定角色。
- 使用 alter role 手动锁定角色：

```
alter role rolename lock
```

Adaptive Server 在 syssrvroles 中包括以下有关锁定信息的列：

- lockdate — 指示何时锁定了角色。
- locksuid — 指示谁锁定了角色。
- lockreason — 给出锁定原因。它采用代码形式：

lockreason 的值	locksuid 的值	角色的 lockreason 的说明
NULL	NULL	角色未锁定
1	alter role 的调用者的 suid	角色已由 suid 通过执行 alter role rolename lock 手动锁定
2	上次尝试角色激活导致角色被锁定的用户的 suid	角色已由 Adaptive Server 锁定，因为角色激活尝试的失败次数达到了 max failed logins。

❖ 锁定和解锁角色

- 系统安全员可使用 alter role 手动锁定或解锁角色。例如：

```
alter role physician_role lock
alter role physician_role unlock
```

有关角色锁定状态的信息存储在 syssrvroles 的 status 列中。

请参见《参考手册：命令》中的 alter role。

注释　在高可用性环境中，这些 syssrvrole 列会在主服务器和辅助服务器上都更新。

锁定拥有阈值的登录名

本节讨论阈值及其如何受锁定的用户登录影响。

- 作为一项安全性措施，可使用帐户名和创建此过程的登录角色来执行阈值存储过程。
  - 不能删除拥有阈值的用户登录。
  - 如果锁定拥有阈值的用户登录，则用户不能执行存储过程。
- 最后机会阈值和使用 “sa” 登录创建的阈值不受 sp_locklogin 影响。如果锁定 “sa” 登录，则由 “sa” 用户创建或修改的最后机会阈值仍会引发。

管理登录配置文件

系统安全员可以定义、改变和删除登录配置文件。

表 3-1 总结了用于创建和管理登录配置文件的系统过程和命令。

表 3-4：在 Adaptive Server 中管理登录配置文件

任务	要求的角色	命令或过程	数据库
创建登录配置文件	系统安全员	create login profile	master 数据库
改变配置文件	系统安全员	alter login profile	master 数据库
删除登录配置文件	系统安全员	drop login profile	master 数据库
返回登录配置文件 ID	系统安全员	lprofile_id	任何数据库
返回登录配置文件名	系统安全员	lprofile_name	任何数据库
显示登录配置文件名	系统安全员	sp_displaylogin	任何数据库
显示有关登录配置文件的信息	系统安全员	sp_securityprofile	任何数据库

登录配置文件属性

表 3-5 总结了登录配置文件的属性。登录配置文件属性存储在 syslogins、sysloginroles 和 master.dbo.sysattributes 中。

表 3-5：登录配置文件属性

属性	说明
default database	Adaptive Server 中的缺省数据库。
default language	缺省语言。
login script	有效的存储过程。通过 create login、 alter login、 create login profile 和 alter login profile 用作登录脚本的存储过程仅限于 120 个字符。
auto activated roles	必须在登录时自动激活的、非口令保护的已授予用户定义角色。如果指定的角色未被授予登录名，则会生成错误。缺省情况下，用户定义的角色不会在登录时自动激活。
authenticate with	指定用于鉴定登录帐户的机制。 如果未指定 authenticate with 鉴定机制，则值 ANY 会用于登录帐户。
track lastlogin	启用上次登录更新。
stale period	指示允许登录帐户在因不活动而被锁定前保持不活动状态的持续时间。
profile id	指定 Adaptive Server 中的数据库。

应用登录配置文件和口令策略属性

可以通过定义登录配置文件作为所有登录帐户、部分登录帐户或单个登录帐户的缺省设置，来管理大量登录帐户的属性。

登录配置文件的属性使用以下优先级与登录帐户相关联：

1　与登录名绑定的登录配置文件的属性值

2　缺省登录配置文件的属性值

3　在以下情况下使用 sp_passwordpolicy 指定的值：

- 缺省登录配置文件不存在
- 还未定义登录配置文件并将其绑定到帐户
- 将登录配置文件设置为忽略 （为命令 create login 指定了参数 with login profile ignore）

4　属性的缺省值

创建登录配置文件

以下步骤讲述如何为特定服务器创建登录配置文件和登录帐户以及为用户管理权限。

1 系统安全员为登录帐户创建登录配置文件。

2 系统安全员为新用户创建登录帐户，并将登录配置文件与新登录帐户相关联。

3 系统管理员或数据库所有者向数据库中添加用户或为组分配用户。

4 系统安全员向用户或登录配置文件授予特定角色。

5 系统管理员、数据库所有者或对象所有者授予用户或组对特定命令和数据库对象的特定权限。

此示例创建登录配置文件 mgr_lp：

```
create login profile mgr_lp
```

请参见 《参考手册：命令》中的 create login profile。

创建缺省登录配置文件

下面的示例创建一个名为 emp_lp 的缺省登录配置文件。如果已有另一个登录配置文件被配置为缺省登录配置文件，则该缺省属性会被删除并应用到 emp_lp：

```
create login profile emp_lp as default
```

请参见 《参考手册：命令》中的 create login profile。

将登录配置文件与登录帐户相关联

如果在创建登录帐户时未指定登录配置文件，则缺省登录配置文件会与新帐户相关联。如果不存在缺省登录配置文件， Adaptive Server 会应用 sp_passwordpolicy 指定的口令策略属性或缺省属性。有关属性应用顺序的信息，请参见应用登录配置文件和口令策略属性。

下面的示例创建口令为 rubaiyat 的登录帐户 omar_khayyam，并将该帐户与登录配置文件 emp_lp 相关联：

```
create login omar_khayyam with password rubaiyat login
      profile emp_lp
```

下面的示例修改登录帐户 omar_khayyam，并将该帐户与登录配置文件 staff_lp 相关联：

```
alter login omar_khayyam modify login profile staff_lp
```

忽略登录配置文件

忽略登录配置文件子句用于禁用直接或通过缺省登录配置文件关联的登录配置文件。 Adaptive Server 采用优先规则来应用登录帐户的对应属性。有关详细信息，请参见应用登录配置文件和口令策略属性。

下面的示例创建一个登录帐户，并指定忽略所有登录配置文件。

```
create login maryb with password itsAsecur8 login
profile ignore
```

将现有的登录帐户值转移到新的登录配置文件

下面的示例演示如何将现有的登录帐户值转移到新的登录配置文件。在创建的登录配置文件 sa_lp 中， default database、 default language 和 authenticate with 属性值设置为和登录帐户 ravi 的值相同。

```
create login profile sa_lp with attributes from ravi
```

手动复制登录配置文件

配置文件 ID 是一个为新登录配置文件指定 ID 的属性，用于在 Adaptive Server 中手动复制登录配置文件。

例如，如果要在复制 master 数据库中创建配置文件 ID 为 25 的配置文件 emp_lp，请执行以下命令：

```
create login profile emp_lp with profile id 25
```

向登录配置文件授予角色

下面的示例创建登录配置文件 def_lp 并向该登录配置文件授予角色 access_role。

```
create login profile def_lp
grant role access_role to def_lp
```

任何绑定到 def_lp 的登录名都将被隐式授予 access_role。系统安全员可以指定授予登录配置文件的角色充当绑定登录名的缺省角色，也就是说，该角色会在用户登录时在其会话中自动激活。

有关添加或删除自动激活的角色，请参见第 60 页的 “添加或删除自动激活的角色”。

调用登录脚本

可以指定登录脚本以便在登录时通过登录配置文件调用。如果通过 sp_logintrigger 指定了全局登录触发器，则会在全局登录触发器之后调用登录脚本。

```
create login profile with login script 'empNew.script'
```

- 可以通过指定登录脚本所在的数据库以及所有者名称来限定登录脚本。如果不用数据库名称限定，则缺省数据库会优先于 master 数据库。
- 如果指定的登录脚本没有用所有者名称限定，则作为当前登录的登录触发器的所有者会优先于登录触发器所在的数据库的所有者。
- 通过 create login、 alter login、 create login profile 和 alter login profile 用作登录脚本的存储过程仅限于 120 个字符。

有关详细信息，请参见第 215 页的 “使用登录触发器”。

显示登录配置文件信息

本节论述如何显示有关登录配置文件的信息。

显示登录配置文件名

若要显示具有指定登录配置文件 ID 或登录 suid 的登录配置文件名，请使用：

lprofile_name([{*login profile id* | *login suid*}])

必须具有系统安全员角色才能查看具有指定登录 ID 的配置文件名 （如果它不是当前用户的登录 ID）。

下面显示具有指定登录配置文件 ID 的登录配置文件名：

```
select lprofile_name(3)
-------------
   intern_lr
```

如果不指定参数，则返回当前用户的登录配置文件名。如果没有登录配置文件与指定的登录帐户相关联，则返回缺省登录配置文件的登录配置文件名。不得设置登录配置文件忽略参数。

还可以使用 sp_displaylogin 显示登录配置文件名。如果没有登录配置文件直接与登录帐户相关联，并且存在缺省登录配置文件，则会显示缺省登录配置文件的名称。

显示登录配置文件 ID

若要显示具有指定登录配置文件名或登录名的登录配置文件 ID，请使用：

*lprofile*_id([{*login profile name* | *login name*}])

必须具有系统安全员角色才能查看具有指定登录名的配置文件 ID （如果它不是当前用户的登录名）。

下面显示具有指定登录配置文件名称的登录配置文件 ID：

```
select lprofile_id('intern_lr')
-------------
   3
```

如果没有登录配置文件与指定的登录帐户相关联，则返回缺省登录配置文件的配置文件 ID。不得设置登录配置文件忽略参数。

显示登录配置文件绑定信息

使用 sp_securityprofile 显示与登录帐户相关联的登录配置文件属性。

注释 非特权登录名只能显示直接与其关联的登录配置文件的属性或缺省登录配置文件的属性。必须具有系统安全员角色才能查看所有登录配置文件的属性和绑定。

有关详细语法信息，请参见 《参考手册：系统过程》中的 sp_securityprofile。

修改登录配置文件

alter login profile 命令可以用于添加、删除或更改登录配置文件的属性及其相应的值。如果未指定这些属性，则它们将添加到登录配置文件中。有关登录配置文件属性的列表，请参见第 55 页的 “登录配置文件属性”。

下面的示例从登录配置文件 mgr_lp 中删除登录脚本属性。如果为缺省登录配置文件指定了登录脚本，则会在登录时调用它，否则将不调用任何登录脚本。

```
alter login profile mgr_lp drop login script
```

有关完整语法，请参见 《参考手册：命令》中的 alter login profile。

添加或删除自动激活的角色

非口令保护的已授予用户定义角色可以在登录时自动激活。

下面修改登录配置文件 mgr_lp，并在与 mgr_lp 关联的用户登录时自动激活角色 mgr_role 和 eng_role。

```
alter login profile mgr_lp add auto activated roles
      mgr_role, eng_role
```

向登录配置文件授予的用户定义角色的自动激活角色状态是在 sysloginroles.status 列中指示的。值为 “1” 指示授予的角色必须在登录时自动激活。撤消某个角色将会删除其在 sysloginroles 中的对应值，该角色将不会在登录时自动激活。 Adaptive Server 如下所示自动激活向用户的登录配置文件授予的角色：

1 如果有缺省登录配置文件与帐户相关联，则会应用在缺省登录配置文件中指定的所有自动激活角色。

2　如果既存在与帐户直接关联的登录配置文件，又存在缺省登录配置文件，则只会应用在与帐户直接关联的登录配置文件中指定的自动激活角色。

将登录配置文件更改为缺省登录配置文件

as [not] 缺省子句用于将登录配置文件指派为缺省登录配置文件或将其作为缺省登录配置文件删除。

下面的语句将名为 emp_lp 的登录配置文件改为缺省登录配置文件。

```
alter login profile emp_lp as default
```

下面的语句将名为 emp_lp 的登录配置文件作为缺省登录配置文件删除。

```
alter login profile userGroup_lp as not default
```

删除登录配置文件

命令 drop login profile 可删除未绑定到登录帐户的登录配置文件。使用 drop login profile with override 可强制删除绑定到登录帐户上的登录配置文件。如果登录配置文件与登录帐户绑定，则登录帐户将绑定到缺省登录帐户 （如果有的话）。如果指定了登录配置文件 ignore 子句，则会删除该子句，并且缺省登录配置文件 （如果有的话）将与登录帐户相关联。

下面的示例强制删除登录配置文件 eng_lp，即使它绑定到一个或多个登录帐户。

```
drop login profile eng_lp with override
```

向数据库添加用户

数据库所有者或系统管理员可以使用 sp_adduser 向特定数据库中添加用户。但用户必须已经拥有 Adaptive Server 登录帐户。语法为：

sp_adduser *loginame* [, *name_in_db* [, *grpname*]]

其中：

- *loginame* — 是现有用户的登录名。
- *name_in_db* — 指定一个与登录名不同的名称，将采用该名称在数据库中标识用户。

使用 *name_in_db* 来适应用户的优先选项。例如，如果有五个名为 Mary 的 Adaptive Server 用户，则每个用户都必须有一个不同的登录名。 Mary Doe 可能以 “maryd” 登录，而 Mary Jones 以 “maryj” 登录，依此类推。但是，如果这些用户没有使用同一数据库，则每个人可能都希望在特定数据库中只以 “mary” 作为登录名。

如果不提供 *name_in_db* 参数，则数据库中的名称与 loginame 相同。

注释 此功能与第 67 页的 “在数据库中使用别名”中描述的别名机制不同，后者将一个用户的标识和权限映射到另一个用户的标识和权限。

- *grpname* — 是数据库中现有组的名称。如果不指定组名称，用户将成为缺省组 “public” 的成员。即使用户是其它组的成员，也仍属于 “public” 组。请参见第 66 页的 “更改用户组成员资格”。

sp_adduser 命令将在当前数据库的 sysusers 系统表中添加一行。当用户在数据库的 sysusers 表中已有一个条目时，用户：

- 可以发出 use *database_name* 命令来访问此数据库
- 将缺省使用此数据库，条件是 create login 指定了缺省数据库参数
- 可以使用 alter login 命令，将此数据库作为缺省数据库

该示例显示数据库所有者如何授予已存在的工程组“eng”中的“maryh”访问权限：

```
sp_adduser maryh, mary, eng
```

下面的示例显示如何授予 “maryd” 访问数据库的权限，并保持其在数据库中的名称与登录名相同：

```
sp_adduser maryd
```

下面的示例显示如何通过使用 null 替代新的用户名，向现有组 “eng” 中添加 “maryj”，并保持其在数据库中的名称与登录名相同：

```
sp_adduser maryj, null, eng
```

可以访问数据库的用户仍需要读取数据、修改数据和使用特定命令的权限。这些权限使用第 6 章 “管理用户权限”中讨论的 grant 和 revoke 命令来授予。

将 “guest” 用户添加到数据库

如果数据库中创建一个名为 “guest” 的用户，就能够使任何拥有 Adaptive Server 帐户的用户作为 **guest** 用户访问该数据库。如果未作为用户或别名用户添加到数据库的用户发出 use *database_name* 命令，Adaptive Server 将查找 guest 用户。如果有，就允许用户以 guest 用户的权限访问数据库。

数据库所有者可以使用 sp_adduser 向数据库的 sysusers 表中添加 guest 条目：

```
sp_adduser guest
```

可以使用第 77 页的 “删除用户”中讨论的 sp_dropuser 命令删除 guest 用户。

如果从 master 数据库中删除 guest 用户，则尚未添加到任何数据库中的服务器用户将无法登录到 Adaptive Server。

注释　尽管可以有多个用户是数据库的 guest 用户，但 Adaptive Server 仍然可以使用用户在服务器内唯一的服务器用户 ID 来审计每个用户的活动。请参见第 8 章 “审计”。

“guest” 用户权限

“guest” 继承 “public” 的特权。数据库所有者和数据库对象的所有者可以使用 grant 和 revoke 使 “guest” 的特权大于或小于 “public” 的特权。请参见第 6 章 “管理用户权限”。

安装 Adaptive Server 时， master..sysusers 中包含一个 guest 条目。

用户数据库中的 “guest” 用户

在用户数据库中，数据库所有者可添加一个允许所有 Adaptive Server 用户使用该数据库的 guest 用户，从而使所有者不必使用 sp_adduser 将每个用户显式命名为数据库用户。

在允许访问数据库时，可以利用 guest 机制限制对数据库对象的访问。

例如， titles 表的所有者可通过执行以下命令，授予除 “guest” 外的所有数据库用户对 titles 的 select 权限：

```
grant select on titles to public
sp_adduser guest
revoke all on titles from guest
```

已安装系统数据库中的 “guest”用户

Adaptive Server 使用 guest 用户创建系统 tempdb 数据库和用户创建的临时数据库。“guest”用户自动拥有在 tempdb 中创建的临时对象和其他对象。 sybsystemprocs、 sybsystemdb 和 sybsyntax 数据库自动包括 “guest”用户。

pubs2 和 *pubs3* 中的 “guest”用户

样本数据库中的 “guest”用户条目允许新的 Adaptive Server 用户按照《Transact-SQL 用户指南》中的示例进行操作。可授予 guest 用户许多特权，包括：

- 对所有用户表的 select 权限与数据修改权限
- 对所有过程的 execute 权限
- create table、 create view、 create rule、 create default 和 create procedure 权限

将 guest 用户添加到服务器

系统安全员可以使用 create login 输入登录名和口令，将指示访问用户使用该登录名和口令。通常只授予这些用户有限的权限。可以指定一个缺省数据库。

警告！ 访问者用户帐户与 “guest”用户帐户不是一回事。所有具有访问者帐户的用户均使用相同的服务器用户 ID；因此不能审计单个用户的活动。每个 “guest”用户都有一个唯一的服务器 ID，因此可审记个人活动并分清个人责任。Sybase 建议您不要创建一个访问者帐户供多个用户使用，因为这样会无法分清个人责任。

可以使用 create login 将名为 “guest”的访问者用户帐户添加到 master..syslogins。此 “guest”用户帐户优先于系统的 “guest”用户帐户。如果用 sp_adduser 添加名为 “guest”的访问者用户，则会影响诸如 sybsystemprocs 和 sybsystemdb 等系统数据库，这些数据库需与数据库中的系统 “guest”用户配合使用。

添加远程用户

可以通过启用远程访问，允许另一个 Adaptive Server 上的用户执行本地服务器上的存储过程。使用远程服务器的系统管理员身份，您还可以允许服务器上的用户执行对远程服务器的**远程过程调用**。

若要启用远程过程调用，您必须对本地服务器和远程服务器均进行重新配置。请参见《系统管理指南：第一卷》中的第 7 章“管理远程服务器”。

创建组

组使您只需通过一个语句即可为多个用户授予或撤消权限，并允许您为一组用户提供一个总称。在管理拥有大量用户的 Adaptive Server 安装时，组非常有用。

请在向数据库中添加用户之前创建组，因为 sp_adduser 既可以将用户指派到组中，也可以将用户添加到数据库中。

您必须具有系统管理员或系统安全员角色或是数据库所有者，才能使用 sp_addgroup 创建组。语法为：

```
sp_addgroup grpname
```

组名 （必需参数）必须遵循标识符规则。系统管理员、系统安全员或数据库所有者可以使用 sp_changegroup 将用户指派或重新指派给组。

例如，若要创建 Senior Engineering 组，请在使用要添加该组的数据库时使用此命令：

```
sp_addgroup senioreng
```

sp_addgroup 命令将在当前数据库的 sysusers 中添加一行。因此，数据库中的每个组以及每个用户在 sysusers 中都有一个条目。

更改用户组成员资格

系统管理员、系统安全员或数据库所有者可以使用 sp_changegroup 更改用户的组从属关系。每个用户都可以只从属于一个不是“public”的组，但所有用户通常都是“public”的成员。

执行 sp_changegroup 之前：

- 组必须存在。
- 用户必须有访问当前数据库的权限（必须在 sysusers 中列出）。

sp_changegroup 的语法为：

```
sp_changegroup grpname, username
```

例如，若要将用户“jim”从当前组更改到“management”组，请使用：

```
sp_changegroup management, jim
```

若要从组中删除一个用户而不将其指派到另一组，必须将其组的从属关系更改为“public”：

```
sp_changegroup "public", jim
```

名称“public”必须用双引号引起，因为它是一个保留字。此命令将 Jim 的组从属关系简化为只属于“public”。

当某个用户从一个组更改到另一个组时，该用户将失去在旧组中拥有的所有权限，而获得向新组授予的权限。

在任何时间都可以更改用户的组指派。

设置组和添加用户

系统安全员、系统管理员或数据库管理员使用 sp_addgroup *group_name* 来创建组。

可以在组级授予和撤消权限。组权限将自动传递到组成员。每个数据库在创建后都有一个名为“public”的组，所有用户均将自动从属于该组。使用 sp_adduser 向组中添加用户，并使用 sp_changegroup 更改用户所在的组。请参见第 66 页的“更改用户组成员资格”。

组由 sysusers 表中的一个条目表示。您不能使用相同的名称在数据库中创建组和用户（例如，不能将某个组和用户同时命名为“shirley”）。

在数据库中使用别名

别名机制允许在数据库中把两个或多个用户当作同一用户对待，因此所有这些用户都具有相同的权限。此机制经常用于使多个用户充当数据库所有者的角色。数据库所有者可以使用 setuser 命令模拟数据库中的另一个用户。也可以使用别名机制设置一个集合用户标识。

例如，假设有几个副总裁要以相同的权限和所有权使用同一个数据库。如果将登录 “vp” 添加到 Adaptive Server 和数据库中，并且让每个副总裁都以 “vp” 登录时，则无法区分开各个用户。反之，若为每个副总裁指定一个数据库用户名 “vp” 的别名，每个人都可以有自己的 Adaptive Server 帐户。

注释　尽管在一个数据库中可能有多人使用别名，但仍可通过审计每个用户所执行的数据库操作来维护个人责任。请参见第 8 章 “审计”。

使用别名过程中的集合用户标识意味着数据库对象的集合所有权。例如，如果用户 “loginA” 在数据库 db1 中的别名为 dbo，则 “loginA” 在 db1 中创建的所有对象由 dbo 拥有。但是， Adaptive Server 将按照登录名和创建者的数据库用户 ID 具体地记录对象的所有权。请参见第 170 页的 “具体标识”。如果别名具体地拥有数据库中的任何对象，则不能将其从该数据库删除。

注释　如果某个登录在数据库中创建了对象，则不能删除该登录的别名。在大多数情况下，只对不拥有表、过程、视图或触发器的用户使用别名。

添加别名

若要为用户添加别名，请使用 sp_addalias：

```
sp_addalias loginame, name_in_db
```

其中：

- *loginame* — 是在当前数据库中需要别名的用户的名称。此用户必须在 Adaptive Server 中有一个帐户且不能是当前数据库的用户。
- *name_in_db* — 是数据库用户的名称，由 *loginame* 指定的用户将被链接到此数据库用户。 *name_in_db* 必须存在于当前数据库的 sysusers 中。

执行 sp_addalias 可将由 *loginame* 指定的用户名映射到由 *name_in_db* 指定的用户名。此功能是通过在系统表 sysalternates 中添加一行实现的。

用户试图使用数据库时，Adaptive Server 将在 *sysusers* 中检查用户的服务器用户 ID 号 (suid)。如果未发现，Adaptive Server 随后检查 sysalternates。如果在此发现用户的 *suid*，并且它已被映射到数据库用户的 *suid*，则第一个用户在其使用数据库时将被看作是第二个用户。

例如，假定 Mary 拥有一个数据库。她希望允许 Jane 和 Sarah 都可以使用此数据库，就像他们是数据库所有者一样。 Jane 和 Sarah 在 Adaptive Server 上有相应的登录名，但未获授权使用 Mary 的数据库。 Mary 执行以下命令：

```
sp_addalias jane, dbo
exec sp_addalias sarah, dbo
```

警告！ 对于讨论中的数据库，具有数据库所有者别名的用户拥有所有权限，并可执行可由数据库所有者执行的所有操作。数据库所有者应仔细考虑授予其他用户数据库全部访问权限可能带来的后果。

删除别名

使用 sp_dropalias 可删除替代 *suid* 与用户 ID 的映射关系。此操作将从 sysalternates 中删除相应的行。语法如下，其中 *loginame* 是在使用 sp_addalias 映射用户名称时由 *loginame* 指定的名称：

sp_dropalias *loginame*

用户的别名被删除后，该用户不再具有访问数据库的权限。

如果别名登录创建了任何对象或阈值，则不能删除此别名。在使用 sp_dropalias 删除已经执行这些操作的别名之前，删除对象或过程。如果删除别名以后仍然要使用它们，应使用不同的所有者来重新创建它们。

获取有关别名的信息

若要显示有关别名的信息，请使用 sp_helpuser。例如，若要找到“dbo”的别名，请执行：

```
sp_helpuser dbo

Users_name      ID_in_db    Group_name    Login_name
----------      --------    ----------    ----------
dbo             1           public        sa

(1 row affected)

Users aliased to user.
Login_name
--------------------
andy
christa
howard
linda
```

获取有关用户的信息

表 3-6 列出了可以用于获取用户、组和当前 Adaptive Server 使用状况相关信息的过程。

表 3-6：报告有关 Adaptive Server 用户和组的信息

任务	过程
报告当前的 Adaptive Server 用户和进程	sp_who
显示有关登录帐户的信息	sp_displaylogin
报告数据库中的用户和别名	sp_helpuser
报告数据库中的组	sp_helpgroup

有关用户和进程的报告

使用 sp_who 报告有关 Adaptive Server 上当前的用户和进程的信息：

sp_who [*loginame* | "*spid*"]

其中：

- *loginame* — 是用户的 Adaptive Server 登录名。如果您提供登录名，则 sp_who 报告有关此用户正在运行的进程的信息。
- *spid* — 是特定进程的编号。

对于每个进程运行， sp_who 会报告服务器进程 ID 的安全相关信息、其状态、进程用户的登录名、真实的登录名 （如果 *login_name* 为别名）、主机计算机的名称、阻塞此进程的进程 （如果存在）的服务器进程 ID、数据库名和正在运行的命令。

如果不提供登录名或 *spid*， sp_who 将报告所有用户正在运行的进程。

下面的示例显示了执行未带参数的 sp_who 命令的安全相关结果：

```
fid spid  status      loginame   origname   hostname          blk_spid  dbname
   tempdbname  cmd               block_xloid    threadpool
--- ----  ---------   ---------  ---------  ---------------   --------  -----------
   ----------- ----------------- -----------    -----------------
  0    1    running         sa         sa            sunbird         0        pubs2
       tempdb  SELECT                     0     syb_default_pool
  0    2   sleeping        NULL       NULL                           0       master
       tempdb  NETWORK HANDLER            0     syb_default_pool
  0    3   sleeping        NULL       NULL                           0       master
       tempdb  MIRROR HANDLER             0     syb_default_pool
  0    4   sleeping        NULL       NULL                           0       master
       tempdb  AUDIT PROCESS              0     syb_default_pool
  0    5   sleeping        NULL       NULL                           0       master
       tempdb  CHECKPOINT SLEEP           0     syb_default_pool
```

sp_who 将所有系统进程的 *loginame* 报告为 NULL。

获取有关登录帐户的信息

使用 sp_displaylogin 可显示有关指定登录帐户或与通配符模式匹配的登录名（包括授予此帐户的任何角色）的信息，其中 *loginame*（或通配符匹配模式）是要获取其有关信息的用户登录名模式：

sp_displaylogin [*loginame* | *wildcard*]

如果您不是系统安全员或系统管理员，则只能显示有关自己的帐户的信息。如果是系统安全员或系统管理员，则可以使用 *loginame* | *wildcard* 参数访问有关任何帐户的信息。

sp_displaylogin 显示您的服务器用户 ID、登录名、全名、授予您的任何角色、上次更改口令的日期、缺省数据库、缺省语言、帐户是否被锁定、任何自动登录脚本、口令有效期、口令是否已经到期、使用的登录口令加密版本，以及为登录指定的鉴定机制。

sp_displaylogin 显示授予用户的所有角色，因此，即使使用 set 命令使某个角色失效时，此角色也将出现。例如，此示例显示 sa 的角色：

```
sp_displaylogin 'sa'

Suid:121
Loginame:mylogin
Fullname:
Default Database:master
Default Language:
Auto Login Script:
Configured Authorization:
        sa_role (default ON)
        sso_role (default ON)
        oper_role (default ON)
        sybase_ts_role (default ON)
Locked:NO
Date of Last Password Change:Aug 10 2006 11:17AM
Password expiration interval:0
Password expired:NO
Minimum password length:6
Maximum failed logins:0
Current failed login attempts:
Authenticate with:NONE
Login password encryption:SYB-PROP, SHA-256
Last login date:Aug 17 2006 5:55PM
 (return status = 0)
```

获取有关数据库用户的信息

使用 sp_helpuser 可报告有关当前数据库的授权用户的信息，其中 *name_in_db* 是当前数据库中的用户名：

sp_helpuser [*name_in_db*]

如果您提供用户名，则 sp_helpuser 将报告有关该用户的信息。如果不提供名称，此命令报告所有用户的有关信息。

下面的示例显示了在 pubs2 数据库中执行未带参数的 sp_helpuser 命令的结果：

```
sp_helpuser
Users_name  ID_in_db  Group_name Login_name
----------  --------  ---------- ----------
dbo         1         public     sa
marcy       4         public     marcy
sandy       3         public     sandy
judy        5         public     judy
linda       6         public     linda
anne        2         public     anne
jim         7         senioreng  jim
```

查找用户名和 ID

若要查找用户的服务器用户 ID 或登录名，请使用 suser_id 和 suser_name。

表 3-7：系统函数 suser_id 和 suser_name

查找	使用	参数
服务器用户 ID	suser_id	(["server_user_name"])
服务器用户名 （登录名）	suser_name	([server_user_ID])

这些系统函数的参数都是可选的。如果不提供参数，则 Adaptive Server 显示有关当前用户的信息。

下面的示例显示了如何查找用户 “sandy”的服务器用户 ID：

```
select suser_id("sandy")

------
     3
```

下面的示例显示登录名为 “mary”的系统管理员如何发出不带参数的命令：

```
select suser_name(), suser_id()
---------------------------- ------
mary                                4
```

若要在数据库内查找用户的 ID 号或名称，请使用 user_id 和 user_name。

表 3-8：系统函数 user_id 和 user_name

查找	使用	参数
用户 ID	user_id	([“db_user_name”])
用户名	user_name	([db_user_ID])

这些函数的参数都是可选的。如果不提供参数，则 Adaptive Server 显示有关当前用户的信息。例如：

```
select user_name(10)
------------------------------------------------------
NULL
(1 row affected)

select user_name( )
------------------------------------------------------
dbo
(1 row affected)

select user_id("joe")
------------------------------------------------------
NULL
(1 row affected)
```

更改用户信息

表 3-9 列出了可用于更改口令、缺省数据库、缺省语言、全名或组指派的系统过程。

表 3-9：用于更改用户信息的命令或系统过程

任务	要求的角色	系统过程	使用的 master 数据库：alter/create/drop login/login profile 的命令
更改用户口令	用户	alter login	任何数据库
更改其他用户的口令	系统安全员	alter login	任何数据库
更改鉴定机制	系统安全员	alter login alter login profile	任何数据库
更改全名	系统安全员	alter login	任何数据库
更改您自己的全名	用户	alter login	任何数据库
更改缺省语言或缺省数据库	系统安全员	alter login profile alter login	任何数据库
更改用户的组指派	系统管理员、数据库所有者或系统安全员	sp_changegroup	用户数据库
更改登录配置文件	系统安全员	alter login profile	任何数据库
配置登录触发器	系统安全员	alter login profile	任何数据库

更改口令

所有用户都使用 alter login 随时更改其口令。系统安全员可以使用 alter login 更改任何用户的口令。

例如，若要更改名为 ron 的登录帐户的口令，请输入：

```
alter login ron with password watsMypaswd modify
password 8itsAsecret
```

请参见 《参考手册：命令》中的 alter login。

要求新口令

可以选择使用 systemwide password expiration 配置参数设定口令有效期，这将强制所有 Adaptive Server 用户定期更改口令。请参见《系统管理指南：第一卷》中的第 5 章 “设置配置参数”。即使不使用 systemwide password expiration，但出于安全原因，用户定期更改口令也是非常重要的。

口令策略设置取代该配置参数。

password expiration interval 指定口令的有效期 （以天为单位）。可以是介于 0 和 32767 之间的任何值 （包括 0 和 32767）。例如，如果您在 2007 年 8 月 1 日上午 10:30 创建一个新登录名，其口令有效期为 30 天，那么该口令将在 2007 年 8 月 31 日上午 10:30 过期。

syslogins 表中的 pwdate 列记录上次更改口令的日期。下面的查询选择自 2007 年 9 月 15 日以来口令未更改的所有登录名：

```
select name, pwdate
from syslogins
where pwdate < "Sep 15 2007"
```

空口令

不要指定一个空口令。安装了 Adaptive Server 之后，缺省 “sa”帐户具有一个空口令。此示例显示如何将空口令更改为有效的口令：

```
alter login sa with password null modify password 8M4LNC
```

注释　在语句中不要将 “null”用引号引起。

丢失口令后登录

如果您的站点遇到以下任意情况，您可以使用 dataserver -p*login_name*：

- 所有系统管理员登录帐户均被锁定。
- 所有系统安全员登录帐户均被锁定。
- sa_role 或 sso_role 的口令已丢失。

利用带 -p 参数的 dataserver 参数，您可以为这些帐户和角色设置新口令。*login_name* 是必须为其重置口令的用户或角色 （sa_role 或 sso_role）的名称。

使用 -p 参数启动时， Adaptive Server 会生成一个随机口令，接着显示并加密该口令，然后将其作为该帐户或角色的新口令保存在 master..syslogins 或 master..syssrvroles 中。

Sybase 强烈建议您在重新启动服务器时更改口令。例如，要为具有 sa_role 的用户 rsmith 重新设置口令，请输入：

```
dataserver -prsmith
```

要重新设置 sso_role 的口令，请输入：

```
dataserver -psso_role
```

更改用户会话信息

set 命令包括的选项可为每个客户端指派各自的名称、主机名和应用程序名。在一个系统中，当多个客户端使用相同的名称、主机名或应用程序名与 Adaptive Server 连接时，该命令对区别这些客户端很有用。

set 命令的部分语法是：

```
set [clientname client_name | clienthostname host_name |
clientapplname application_name]
```

其中：

- *client_name* — 是指派给客户端的名称。
- *host_name* — 是客户端所连接到的主机的名称。
- *application_name* — 与是 Adaptive Server 连接的应用程序。

这些参数存储在 sysprocesses 表的 clientname、 clienthostname 和 clientapplname 列中。

例如，如果用户以 “client1” 登录到 Adaptive Server，则可以使用类似下面的命令为其指派客户机名、主机名和应用程序名：

```
set clientname 'alison'
set clienthostname 'money1'
set clientapplname 'webserver2'
```

此用户此时作为用户 “alison” 从主机 “money1” 登录并使用 “webserver2” 应用程序，显示在 sysprocesses 表中。然而，尽管新名称出现在 sysprocesses 中，但它们并没有用于权限检查，且 sp_who 仍将客户端连接显示为属于初始登录 （在上述情况中为 client1）。 set clientname 不执行与 set proxy 相同的功能，后者允许用户使用另一个用户的权限、登录名和 *suid*。

可以为唯一当前客户端会话设置客户端名、主机名或应用程序名 （尽管可以查看任何客户端连接的连接信息）。此外，此信息在用户注销时将丢失。每次用户登录时，这些参数必须都要重新指定。例如，用户 “alison” 不能为任何其它客户端连接设置客户端名、主机名或应用程序名。

使用客户端的系统进程 ID 可以查看其连接信息。例如，如果上面提到的用户 “alison” 使用 *spid* 13 进行连接，则发出下面的命令可查看此用户的所有连接信息：

```
select * from sysprocesses where spid = 13
```

若要查看当前客户端连接的连接信息 （例如，如果用户 “alison” 要查看自己的连接信息），请输入：

```
select * from sysprocesses where spid = @@spid
```

删除用户和组

系统管理员、系统安全员或数据库所有者可以使用 sp_dropuser 或 sp_dropgroup 从数据库中删除用户和组。

删除用户

数据库所有者、系统安全员或系统管理员可以使用 sp_dropuser，拒绝 Adaptive Server 用户访问执行 sp_dropuser 的数据库。（如果在此数据库中定义了 “guest”用户，则该用户仍可以作为 “guest”访问该数据库。）

以下是语法，其中 *name_in_db* 通常是登录名，除非使用 sp_adduser 指派了另一个名称：

```
sp_dropuser name_in_db
```

不能删除拥有对象的用户。由于没有用来转移对象所有权的命令，因此在删除用户之前，必须先删除用户所拥有的对象。若要拒绝对拥有对象的用户进行访问，请使用 sp_locklogin 锁定其帐户。

也不能删除已授予其他用户权限的用户。可使用 revoke with cascade 命令先撤消所有由要删除的用户授予了权限的用户的权限，然后删除该用户。如果需要，必须再次为这些用户授予权限。

删除组

系统安全员、系统管理员或数据库管理员使用 sp_dropgroup 来删除组。语法为：

```
sp_dropgroup grpname
```

不能删除包含成员的组。如果试图这样做，错误报告将显示试图删除的组中的成员列表。若要从组中删除用户，请使用第 66 页的 “更改用户组成员资格”中讨论的 sp_changegroup。

监控许可证的使用状况

系统管理员可以通过许可证使用监控器监控在 Adaptive Server 中使用的用户许可证数，同时安全地管理许可协议数据。这就是说，可以确保 Adaptive Server 中使用的许可证数没有超过许可协议中指定的数量。

许可证使用监控器追踪执行的许可证数；但不强制执行许可协议。如果许可证使用监控器报告正在使用的用户许可证数超出了许可协议中指定的数量，请询问 Sybase 销售代表。

您必须具有系统管理员特权才能配置许可证使用监控器；缺省情况下，在安装或升级 Adaptive Server 时，该监控器处于禁用状态。

请参见下面的 “配置许可证使用监控器”。

如何计算许可证数

许可证是指主机名和用户名的组合。如果用户从同一主机多次登录到 Adaptive Server，将使用一个许可证。然而，如果用户从主机 A 登录一次，从主机 B 登录一次，则使用两个许可证。如果多个用户以不同的用户名从同一主机登录到 Adaptive Server，则用户名与主机名的每个不同组合都使用一个许可证。

配置许可证使用监控器

使用 sp_configure 指定许可协议中的许可证数，其中 *number* 是许可证数：

```
sp_configure "license information" , number
```

此示例将用户许可证的最大数目设置为 300，如果许可证数目为 301，则报告出现了过度使用：

```
sp_configure "license information", 300
```

如果增加用户许可证的数目，还必须更改 license information 配置参数。

通过管家任务监控许可证使用状况

配置许可证使用监控器后，管家任务根据登录到 Adaptive Server 的每个用户的用户 ID 和主机名，来确定在使用中的用户许可证数。许可证使用监控器更新一个变量，该变量跟踪正在使用的用户许可证的最大数目：

- 如果使用中的许可证数自上次运行管家任务以来没有变化或已减少，许可证使用监控器将不进行任何操作。
- 如果在使用的许可证数自上次管家运行后已经添加，许可证使用监控器则将此数目设置为在使用的许可证最大数目。
- 如果在使用的许可证数大于许可协议允许的数目，则许可证使用监控器将此消息发送到错误日志：

  超过许可证使用限制。请与 Sybase 销售部门联系以获取额外的许可证。

管家杂事任务在 Adaptive Server 空闲周期运行。 housekeeper free write percent 和 license information 配置参数均必须设置为大于或等于 1 的值，以便许可证使用监控器跟踪许可证使用情况。

有关管家杂事任务的详细信息，请参见 Performance and Tuning Series: Basics （《性能和调优系列：基础知识》）中的第 3 章 “Using Engines and CPUs”（使用引擎和 CPU）。

记录用户许可证数

syblicenseslog 系统表是在安装或升级 Adaptive Server 时在 master 数据库中创建的。许可证使用监控器在每个 24 小时周期结束时，更新 syblicenseslog 中的列，如表 3-10 所示。

表 3-10：syblicenseslog 表中的列

列	说明
status	-1 — 管家不能监控许可证。 0 — 没有超过许可证数量。 1 — 超过了许可证数量。
logtime	插入日志信息的日期和时间。
maxlicenses	在前 24 小时内使用的最大许可证数。

syblicenseslog 形式类似如下所示：

```
status logdate                    maxlicenses
------ ------------------------ -----------
     0    Jul 17 1998 11:43AM          123
     0    Jul 18 1998 11:47AM          147
     1    Jul 19 1998 11:51AM          154
     0    Jul 20 1998 11:55AM          142
     0    Jul 21 1998 11:58AM          138
     0    Jul 21 1998  3:14PM          133
```

在本例中，在 1998 年 7 月 19 日使用的用户许可证数超过了限制。

如果 Adaptive Server 关闭，则许可证使用监控器会以所用许可证当前的最大数目来更新 syblicenseslog。当 Adaptive Server 重新启动时，开始新的 24 小时监控周期。

1998 年 7 月 21 日记录的第二行是由于服务器的关机和重新启动引起。

用户数和登录 ID 数

Adaptive Server 支持每个服务器有超过 2,000,000,000 个登录名，每个数据库有超过 2,000,000,000 个用户。 Adaptive Server 可使用负数和正数增加可用 ID 的范围。

ID 号的限制和范围

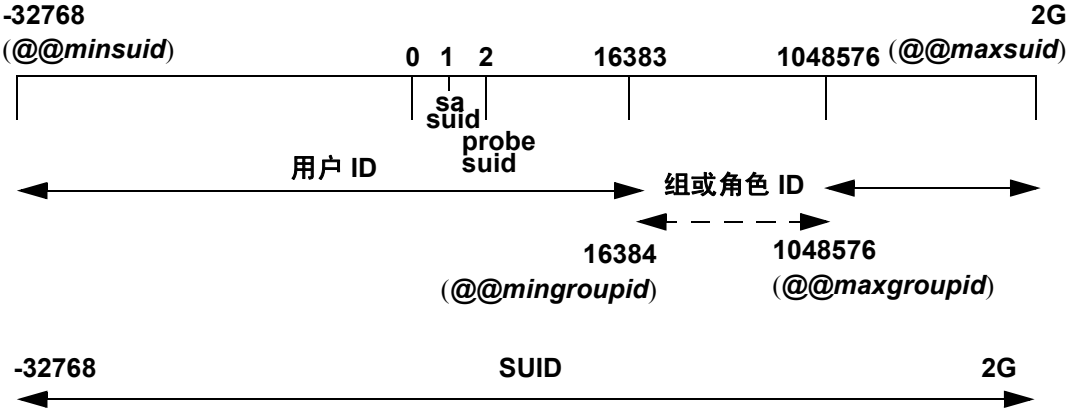
表 3-11 说明了 ID 类型的有效范围。

表 3-11：ID 类型的范围

ID 类型	服务器限制
每个服务器的登录数 (*suid*)	2G 加上 32K
每个数据库的用户数 (*uid*)	2G 减去 1032193
每个数据库的组数或角色数 (*gid*)	16,384 到 1,048,576

图 3-1 说明了登录名、用户和组的限制和范围。

图 3-1：Adaptive Server 中可用的用户、组和登录名



可以为用户 ID (*uid*) 使用负值。

与 sysusers 中一个组或角色关联的服务器用户 ID (*suid*) 不等于其用户 ID (*uid*) 的负值。在 *sysusers* 中，将每个与组或角色相关联的 suid 设置为 -2 (INVALID_SUID)。

登录连接限制

尽管 Adaptive Server 允许为每个服务器定义超过 2,000,000,000 个登录名，但可以同时连接到 Adaptive Server 的实际用户数受以下因素限制：

- number of user connections 配置参数的值和
- 可用于 Adaptive Server 的文件描述符数。每个登录都使用一个文件描述符用于连接。

注释　服务器上运行的最大并发任务数为 32,000。

❖ 允许登录和同时连接的最大值

1　为运行 Adaptive Server 的操作系统配置至少 32,000 个文件描述符。

2　将 number of user connections 的值至少设置为 32,000。

注释　在 Adaptive Server 可以处理数量超过 64K 的登录和同时连接之前，必须先将操作系统配置为可以使用 64K 以上的文件描述符。有关增加文件描述符数的信息，请参见操作系统文档。

表 3-12：登录、用户和组的全局变量

变量名称	显示内容	值
@@invaliduserid	无效用户 ID	-1
@@minuserid	最小的用户 ID	-32768
@@guestuserid	Guest 用户 ID	2
@@mingroupid	最小的组或角色用户 ID	16384
@@maxgroupid	最大的组或角色用户 ID	1048576
@@maxuserid	最大的用户 ID	2147483647
@@minsuid	最小的服务器用户 ID	-32768
@@probesuid	Probe 服务器用户 ID	2
@@maxsuid	最大的服务器用户 ID	2147483647

若要发出全局变量，请输入：

```
select variable_name
```

例如：

```
select @@minuserid
-----------
-32768
```

获取有关使用情况的信息：收费退回式会计

用户登录到 Adaptive Server 时，服务器开始累计此用户的 CPU 和 I/O 使用状况。Adaptive Server 可以报告单个用户或所有用户的总体使用状况。每个用户的信息都存储在 master 数据库的 syslogins 系统表中。

报告当前使用状况统计信息

系统管理员可以使用 sp_reportstats 或 sp_clearstats，获取或清除有关 Adaptive Server 上单个用户或所有用户的当前总体使用状况的数据。

显示当前会计汇总

sp_reportstats 显示 Adaptive Server 用户的当前会计汇总。此命令报告全部 CPU 和 I/O 时间，以及使用这些资源的百分比。但不记录“sa”登录名 （*suid* 为 1 的进程）、检查点、网络和镜像处理程序的统计信息。

初始化新的会计间隔

Adaptive Server 将累计 CPU 和 I/O 时间，直到通过运行 sp_clearstats 从 syslogins 中清除汇总数据。sp_clearstats 为 Adaptive Server 用户启动一个新的会计间隔，并执行 sp_reportstats 以打印出上一个时间段的统计信息。

通过确定如何在自己的节点使用统计信息来选择会计间隔的时间长度。例如，若要每月跨部门按 Adaptive Server 的 CPU 和 I/O 使用百分比收取费用，可每月运行一次 sp_clearstats。

有关这些存储过程的详细信息，请参见 《参考手册：过程》。

指定添加会计统计信息的间隔

系统管理员可以使用配置参数确定将会计统计信息添加到 syslogins 的频率。

若要指定在会计统计信息被添加到 syslogins 之前计算机时钟累计的周期数，请使用 cpu accounting flush interval 配置参数。缺省值为 200。例如：

```
sp_configure "cpu accounting flush interval", 600
```

若要获取系统上一个时钟周期的微秒数，可在 Adaptive Server 中运行下面的查询：

```
select @@timeticks
```

若要指定在信息被添加 （刷新）到 syslogins 之前读取或写入 I/O 的累计次数，请使用 i/o accounting flush interval 配置参数。缺省值为 1000。例如：

```
sp_configure "i/o accounting flush interval", 2000
```

I/O 和 CPU 统计信息在用户的累计 I/O 或 CPU 使用超过指定值时刷新。这些信息在用户退出 Adaptive Server 会话时也被刷新。

以上两个配置参数所允许的最小值为 1，最大值为 2,147,483,647。

第 4 章

外部鉴定

本章介绍了某些 Adaptive Server 功能，通过这些功能可以用存储在 Adaptive Server 外部存储库中的鉴定数据来鉴定用户。

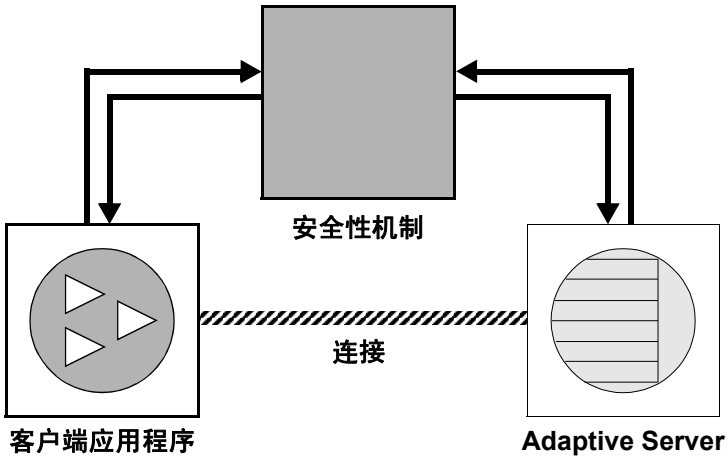
通过使用中央存储库鉴定登录名，可以增强大型异构应用程序的安全性。 Adaptive Server 支持以下外部鉴定方法：

- Kerberos — 可在使用 Kerberos 基础结构的企业环境中，提供一种集中且安全的鉴定机制。使用一台名为密钥分发中心 (KDC) 的第三方受托服务器进行鉴定，以同时验证客户端和服务器。
- LDAP 用户鉴定 — 轻量目录访问协议 (LDAP) 可根据用户的登录名和口令，提供一种集中式鉴定机制。
- PAM 用户鉴定 — 可插入鉴定模块 (PAM) 可以同时为管理和运行时应用程序界面，提供一种集中式鉴定机制，该机制可使用操作系统提供的界面。

配置 Adaptive Server 以实现基于网络的安全性

客户端和服务器之间的安全连接可用于登录鉴定和消息保护。

图 4-1：在客户端和 Adaptive Server 之间建立安全连接



如果客户端请求验证服务：

1 客户端使用安全性机制验证登录。安全性机制返回一个包含安全性相关信息的认证。

2 客户端将认证发送到 Adaptive Server。

3 Adaptive Server 使用安全性机制鉴定客户端的认证。如果凭据有效，在客户端和 Adaptive Server 之间建立安全连接。

如果客户端请求消息保护服务：

1 客户端使用安全性机制来准备将发送到 Adaptive Server 的数据包。

根据所请求的安全服务，安全性机制可能会加密数据或创建一个与数据关联的密码签名。

2 客户端将数据包发送到 Adaptive Server。

3 当收到数据包时， Adaptive Server 将使用安全性机制执行任何必需的解密和验证操作。

4 Adaptive Server 将结果返回给客户端，并使用安全性机制执行所请求安全性功能。例如， Adaptive Server 可能以加密的形式返回结果。

安全服务和 Adaptive Server

根据所选的安全性机制， Adaptive Server 允许您使用以下的一个或多个安全服务：

- 统一登录 — 一次性鉴定用户，而不要求用户每次登录到 Adaptive Server 时都提供用户名和口令。
- 消息保密性 — 加密网络上的数据。
- 相互鉴定 — 验证客户端和服务器的标识。相互鉴定只能由客户端请求；而不能由 Adaptive Server 要求。
- 消息完整性 — 验证数据通信是否未被修改。
- 重放检测 — 验证数据是否被入侵者截取。
- 顺序混乱检查 — 验证验数据通信的顺序。
- 消息源检查 — 验证消息源。
- 凭据委托 — 能让客户端将凭据委托给 Adaptive 服务器，以便启用与远程服务器的安全连接。该服务由 Kerberos 安全性机制支持。 Adaptive 服务器当前支持该服务通过 CIS 连接到远程 Adaptive 服务器。
- 远程过程安全性 — 为通过 CIS 实现 Kerberos 连接的远程过程通信建立相互鉴定、消息保密性和消息完整性。

注释　正在使用的安全性机制可能无法使用上述所有服务。请参见第 101 页的 “获取有关可用安全服务的信息”。

管理基于网络的安全性

表 4-1 提供了使用基于网络的安全性功能 （由 Adaptive Server 提供）的全部过程。必须先安装 Adaptive Server，然后才能完成表 4-1 中的步骤。

表 4-1：管理基于网络的安全性

步骤	说明	请参见
1. 设置配置文件： • *libtcl.cfg* • *objectid.dat* • *interfaces* （或目录服务）	编辑 *libtcl.cfg* 文件。 编辑 *objectid.dat* 文件。 编辑 *interfaces* 文件或目录服务。	• 第 89 页的 “为安全性设置配置文件” • 所用平台的 Open Client/Server Configuration Guide （《Open Client/Server 配置指南》）
2. 确保安全性机制的安全管理员已为每个用户及 Adaptive Server 和 Backup Server 创建了登录。	安全管理员必须为安全性机制中的用户和服务器添加名称和口令。	• 随安全性机制提供的文档 • 第 93 页的 “向安全性机制标识用户和服务器”
3. 配置所安装产品的安全性。	使用 sp_configure。	第 94 页的 “配置 Adaptive Server 的安全性”
4. 重新启动 Adaptive Server。	激活 use security services 参数。	所用平台的 《配置指南》
5. 将登录添加到 Adaptive Server 以支持企业范围的登录。	使用 create login 添加登录帐户。另外，还可以用 sp_configure 指定缺省安全登录。	第 97 页的 “添加登录以支持统一登录”
6. 为必需的远程服务器启用安全性机制。	使用 sp_serveroption 的 security mechanism 选项为必需的远程服务器启用安全性机制。	第 98 页的 “为远程连接建立 Kerberos 安全机制”
7. 连接到服务器并使用安全服务。	使用 isql_r 或 Open Client Client-Library 连接到 Adaptive Server，并指定要使用的安全服务。	• 第 100 页的 “连接到服务器并使用安全服务” • 所用平台的 Open Client/Server Configuration Guide （《Open Client/Server 配置指南》） • 《Open Client Client-Library/C 参考手册》中的 “安全性功能”
8. 检查可用的安全服务及安全性机制。	使用函数 show_sec_services 和 is_sec_services_on 检查哪种安全服务可用。 有关 Adaptive Server 所支持的一组安全性机制及其安全服务，可使用 select 来查询 syssecmechs 系统表。	第 101 页的 “获取有关可用安全服务的信息”

为安全性设置配置文件

在安装过程中，会在 Sybase 目录结构的缺省位置创建配置文件。

表 4-2：配置文件的名称和位置

文件名	说明	位置
libtcl.cfg	驱动程序配置文件包含有关目录、安全性和网络驱动程序的信息，还包含任何所需的初始化信息。	*UNIX 平台*：*$SYBASE/$SYBASE_OCS/config* *Windows 平台*：*%SYBASE%\%SYBASE_OCS%\ini*
objectid.dat	此对象标识符文件将全局对象标识符映射为字符集、归类序列和安全性机制的本地名。	*UNIX 平台*：*$SYBASE/config* *Windows 平台*：*%SYBASE%\ini*
UNIX：*interfaces* *桌面平台*：*sql.ini*	*interfaces* 文件包含关于文件中所列出的每个服务器的连接和安全性信息。 **注释**　在 Adaptive Server 12.5.1 及更高版本中，可以使用目录服务代替 *interfaces* 文件。	*UNIX 平台*：*$SYBASE* *桌面平台*：*SYBASE_home\ini*

有关配置文件的详细说明，请参见所用平台的 Open Client/Server Configuration Guide （《Open Client/Server 配置指南》）。

指定服务器的安全信息

使用 *interfaces* 文件或目录服务，以提供有关所安装服务器的信息。

interfaces 文件包含服务器的网络及安全性信息。若要使用安全服务，*interfaces* 文件必须包括 “secmech” 行，该行指定全局标识符或计划使用的安全服务的标识符。

Adaptive Server 支持目录服务以跟踪有关服务器的信息。目录服务管理网络服务器信息的创建、修改和检索。使用目录服务的好处在于，当网络上新添加服务器或者服务器移动到新地址时，无须更新多个 *interfaces* 文件。若要将安全服务与目录服务一起使用，您必须定义 secmech 安全属性，以指向一个或多个计划使用的安全服务的全局标识符。

指定安全性机制的 UNIX 工具

要定义安全性机制，请执行以下操作：

- 如果使用的是 *interfaces* 文件，请使用 dscp 实用程序。
- 如果使用的是目录服务，则运行 dscp_r 实用程序。

注释　UNIX 平台上提供的 dsedit 工具有助于创建 *interfaces* 文件或目录服务的条目。然而，它无法支持创建安全性机制的 secmech 条目。

有关 dscp 的详细信息，请参见 《用于 UNIX 的 Open Client/Server 配置指南》。

用于指定服务器属性的桌面工具

若要在 sql.ini 文件或目录服务中提供所安装的服务器的信息，请使用 dsedit 实用程序。此实用程序提供一个图形用户界面，用来指定服务器属性（如服务器版本、名称和安全性机制）。对于安全性机制属性，可以为计划使用的安全性机制指定一个或多个对象标识符。有关使用 dsedit 的信息，请参见 Open Client/Server Configuration Guide for Desktop Platforms（《用于桌面平台的 Open Client/Server 配置指南》）。

准备 *libtcl.cfg* 以使用基于网络的安全性

libtcl.cfg 和 *libtcl64.cfg*（用于 64 位应用程序）包含有关三种类型驱动程序的信息：

- 网络 (Net-Library)
- 目录服务
- 安全性

驱动程序是一个 Sybase 库，可提供到外部服务提供程序的接口。驱动程序可动态装载，因而，在更改某个应用程序使用的驱动程序时无需重新链接到该应用程序。

网络驱动程序条目

网络驱动程序条目的语法是：

driver=protocol description

其中：

- *driver* — 网络驱动程序的名称。
- *protocol* — 网络协议的名称。
- *description* — 条目说明。此元素可选。

注释 如果不指定网络驱动程序，将自动使用适合当前应用程序和平台的驱动程序。例如，对于 UNIX 平台，在使用安全服务时，会自动选用可以处理线程的驱动程序。

目录服务条目

如果要使用目录服务而不是 *interfaces* 文件，则目录服务条目适用。请参见所用平台的配置文档，以及所用平台的 Open Client/Server Configuration Guide （《Open Client/Server 配置指南》）。

安全性驱动程序条目

安全性驱动程序条目的语法为：

provider=driver init-string

其中：

- *provider* — 是安全性机制的本地名。在 *objectid.dat* 中定义本地名到全局对象标识符的映射。

  缺省的本地名是：

  - “csfkrb5” — CyberSAFE 或 MIT Kerberos 的安全性机制。
  - “LIBSMSSP” — 用于 Windows NT 或 Windows 95 平台 （仅适用于客户端）上的 Windows LAN Manager。

  如果使用非缺省的本地机制名称，请在 objectid.dat 文件中更改相应的本地名 （例如，请参见第 92 页的 “objectid.dat 文件”）。

- *driver* — 安全性驱动程序的名称。用于 UNIX 平台的所有驱动程序的缺省位置为 *$SYBASE/$SYBASE_OCS/lib*。驱动程序在 Windows 平台的缺省位置为 *%SYBASE%\%SYBASE_OCS%\dll*。

- *init-string* — 驱动程序的初始化字符串。此元素可选。不同驱动程序的 *init-string* 值各不相同：

  - Kerberos 驱动程序 — 以下是 *init-string* 的语法，其中 *realm* 是缺省的 Kerberos 领域名：

    secbase=@*realm*

  - Windows NT LAN Manager — *init-string* 不适用。

UNIX 平台信息

没有可用于编辑 *libtcl.cfg* 文件的特殊工具。使用您所喜欢的编辑器，注释掉那些在安装 Adaptive Server 后已经存在的条目，或撤消对这些条目的注释。

您在 UNIX 平台上安装 Adaptive Server 后， *libtcl.cfg* 文件已包含对应于文件以下三个部分的条目：

- [DRIVERS]
- [DIRECTORY]
- [SECURITY]

各部分无需依照特定顺序排列。

确保注释那些不想使用的条目 （以 “;” 开头），而取消对那些要使用的条目的注释 （不以 “;” 开头）。

有关详细信息，请参见 Open Client/Server Configuration Guide for UNIX （《用于 UNIX 的 Open Client/Server 配置指南》）。

用于 Sun Solaris 的 *libtcl.cfg* 样本

```
[DRIVERS]
;libtli.so=tcp unused ; This is the non-threaded tli driver.
;libtli_r.so=tcp unused ; This is the threaded tli driver.

[SECURITY]
csfkrb5=libsybskrb.so secbase=@MYREALM libgss=/krb5/lib/libgss.so
```

此文件不使用目录服务，因为所有 [DIRECTORY] 部分的条目均已被注释。

由于 [DRIVERS] 部分中的所有网络驱动程序条目也已被注释，因此系统会自动选择合适的驱动程序。当您使用安全服务时， Adaptive Server 会自动选择线程驱动程序，而对于那些不能使用线程驱动程序的应用程序，则会自动选择非线程驱动程序。例如， Backup Server 不支持安全服务，不使用线程驱动程序。

桌面平台信息

ocscfg 实用程序自动为 *libtcl.cfg* 文件创建节标题；您也可以使用 osccfg 来编辑 *libtcl.cfg* 文件。

以下是用于桌面平台的 *libtcl.cfg* 样本文件：

```
[NT_DIRECTORY]
ntreg_dsa=LIBDREG  ditbase=software\sybase\serverdsa

[DRIVERS]
NLWNSCK=TCP  Winsock TCP/IP Net-Lib driver
NLMSNMP=NAMEPIPE  Named Pipe Net-Lib driver
NLNWLINK=SPX  NT NWLINK SPX/IPX Net-Lib driver
NLDECNET=DECNET  DecNET Net-Lib driver

[SECURITY]
NTLM=LIBSMSSP
```

请参见 Open Client/Server Configuration Guide for Desktop Platforms（《用于桌面平台的 Open Client/Server 配置指南》）。

objectid.dat 文件

objectid.dat 文件将全局对象标识符 （例如用于 Kerberos 服务的某个标识符，举例来说，诸如 1.3.6.1.4.1.897.4.6.6 之类的标识符）映射到本地名（例如，“csfkrb5”）。 objectid.dat 文件包括多个部分，例如用于字符集的 [CHARSET] 和用于安全服务的 [SECURITY]。以下是 objectid.dat 样本文件：

```
secmech]
       1.3.6.1.4.1.897.4.6.3   = NTLM
       1.3.6.1.4.1.897.4.6.6   = csfkrb5
```

只有在更改了 *libtcl.cfg* 文件中安全服务的本地名后，才可使用文本编辑器更改此文件。

例如，如果将

```
[SECURITY]
csfkrb5=libsybskrb.so secbase=@MYREALM
libgss=/krb5/lib/libgss.so
```

更改为：

```
[SECURITY]
csfkrb5_group=libsybskrb.so secbase=@MYREALM
libgss=/krb5/lib/libgss.so
```

更改 *libtcl.cfg* 中的 objectid.dat 以反映更改。只需在 objectid.dat 中，更改 Kerberos 行的本地名即可：

```
1.3.6.1.4.1.897.4.6.6 = csfkrb5_group
```

注释　只能为每个安全性机制指定一个本地名。

向安全性机制标识用户和服务器

安全性机制的安全管理员必须为安全性机制定义主管 （用户和服务器）。可用于添加用户和服务器的工具有：

- Kerberos — 有关定义用户和服务器的信息，请参见 Kerberos 供应商特定的工具。有关 Kerberos 和 Adaptive Server 的详细信息，请参见第 102 页的 “使用 Kerberos”。
- Windows NT LAN Manager — 运行 “用户管理器” 工具定义 Windows NT LAN Manager 的用户。将 Adaptive Server 名称作为用户定义到 Windows NT LAN Manager 中，并将 Adaptive Server 显示为该用户名。

注释　在生产环境中，请控制对包含服务器和用户键的文件的访问。如果用户可以访问这些键，他们就能创建一个服务器来模拟您的服务器。

有关如何执行所需管理任务的详细信息，请参见第三方提供程序提供的安全性机制文档。

配置 Adaptive Server 的安全性

Adaptive Server 包括几个用于管理基于网络的安全性的配置参数。必须是系统安全员才能设置这些参数。用于基于网络的安全性的全部参数都是 “安全性相关”配置参数组的一部分。

启用基于网络的安全性

若要启用或禁用基于网络的安全性，请使用 sp_configure 来设置 use security services 配置参数。

如果 use security services 被设置为 1，则以下两个条件同时成立时，Adaptive Server 支持安全性机制：

- *interfaces* 文件或目录服务中列出了安全性机制的全局标识符。
- 在 *objectid.dat* 中，全局标识符映射到列在 *libtcl.cfg* 中的本地名。

有关 Adaptive Server 如何确定用于特定客户端的安全性机制的信息，请参见第 101 页的 “使用用于客户端的安全性机制”。

要求统一登录

若要要求安全性机制监控除系统安全员外的所有用户，请将 unified login required 配置参数设置为 1。设置了此配置参数后，只有具有 sso_role 的用户才能使用用户名和口令登录到服务器：

```
sp_configure "unified login required", [0|1]
```

例如，要求一个安全性机制鉴定所有登录，需执行：

```
sp_configure "unified login required", 1
```

建立安全缺省登录

当一个具有有效的安全性机制认证的用户登录到 Adaptive Server 时，服务器将检查在 master..syslogins 中是否存在该用户名。如果存在，则 Adaptive Server 使用该用户名。例如，如果用户以 “ralph”身份登录到 Kerberos 安全性机制中且在 master..syslogins 中存在 “ralph”，则 Adaptive Server 将使用在服务器中为 “ralph”定义的所有角色和授权。

但是，如果具有有效凭据的用户登录到 Adaptive Server，但该服务器却不知道该用户，则仅当使用 sp_configure 定义了安全缺省登录的情况下，才接受此登录名。 Adaptive Server 对未在 master..syslogins 中定义但已预先通过安全性机制鉴定的所有用户使用缺省登录名。语法为：

```
sp_configure "secure default login", 0, login_name
```

secure default login 的缺省值为 “guest”。

安全缺省登录名必须是 master..syslogins 中的有效登录名。例如，若要将 “gen_auth” 设置为缺省登录名：

1　使用 create login 将添加此登录名作为 Adaptive Server 中的有效用户：

```
create login gen_auth with password pwgenau
```

此过程将初始口令设置为 “pwgenau”。

2　指定该登录作为安全缺省值：

```
sp_configure "secure default login", 0, gen_auth
```

Adaptive Server 对预先通过安全性机制鉴定但 Adaptive Server 不知道的用户使用此登录。

注释　多个用户可以使用与安全缺省登录名关联的 suid。因此，可能需要为缺省登录的全部活动激活审计。可能还要考虑使用 create loign 来向服务器添加用户。

请参见第 15 页的 “创建登录帐户”。

将安全性机制登录名映射到服务器名

某些安全性机制可能允许在 Adaptive Server 中无效的登录名。例如，长度超过 30 个字符的登录名或包含特殊字符（例如 !、%、* 和 &）的登录名在 Adaptive Server 中无效。 Adaptive Server 中的所有登录名都必须是有效标识符。请参见 《参考手册》中的第 3 章 “表达式、标识符和通配符”。

表 4-3 显示了 Adaptive Server 如何转换登录名中的无效字符：

表 4-3：登录名中无效字符的转换

无效字符	转换为
与符号 & 撇号 ’ 反斜杠 \ 冒号 : 逗号 , 等号 = 左引号 ‘ 百分号 % 右尖括号 > 右引号 ’ 变音符号 ~	下划线 _
补注号 ^ 大括号 { } 感叹号 ! 左尖括号 < 小括号 ( ) 句号 。 问号 ?	美元符号 $
星号 * 减号 - 竖线 | 加号 + 引号 " 分号 ; 斜杠 / 方括号 [ ]	# 符号

要求使用加密的消息保密性

若要要求对进出 Adaptive Server 的所有消息进行加密，请将 msg confidentiality reqd 配置参数设置为 1。如果此参数为 0 （缺省值），则不要求消息的保密性，但客户端仍可以建立消息保密性。语法为：

```
sp_configure configuration_parameter, [0 | 1]
```

例如，若要求所有消息被加密，需执行：

```
sp_configure "msg confidentiality reqd", 1
```

要求数据完整性

在 Adaptive Server 中，您可以使用 msg integrity reqd 配置参数，来要求检查所有消息中的一种或多种类型数据的完整性。将 msg integrity reqd 设置为 1，以要求检查所有消息的常规篡改。如果 msg integrity reqd 为 0（缺省值），则不要求消息完整性，但客户端可以建立消息完整性（如果安全性机制支持）。

基于网络安全性的内存要求

为每个安全连接分配大约 2K 的附加内存。max total_memory 配置参数的值指定 Adaptive Server 启动时需要的内存量。例如，如果服务器使用 2K 逻辑页，且如果预计同时发生的安全连接最大数目是 150，请将 max total_memory 参数增加 150，这使得内存分配增加了 150 个 2K 块。

语法为：

sp_configure “max total_memory”, *value*

例如，如果 Adaptive Server 要求 75,000 个 2K 的内存块（包括增加的用于基于网络安全性的内存），需执行：

```
sp_configure "max total_memory", 75000
```

请参见《系统管理指南：卷 2》中的第 4 章“配置数据高速缓存”。

添加登录以支持统一登录

当用户使用预鉴定凭据登录到 Adaptive Server 时， Adaptive Server 将执行以下操作：

1　检查该用户是否为 master..syslogins 中的有效用户。如果用户被列在 master..syslogins 中，则 Adaptive Server 不需要口令就接受此登录。

2　如果用户名不在 master..syslogins 中，则 Adaptive Server 检查是否定义了缺省安全登录。如果定义了缺省登录，则该用户将使用缺省值成功登录。如果未定义缺省登录，则用户无法登录。

因此，需要考虑是只允许被定义为有效登录的用户使用 Adaptive Server，还是希望用户能够使用缺省登录进行登录。若要定义缺省值，请在 master..syslogins 中添加缺省登录，并使用 sp_configure。请参见第 94 页的“建立安全缺省登录”。

添加登录的常规过程

按照表 4-4 中介绍的常规过程将登录添加到服务器，并选择将具有相应角色和授权的用户添加到一个或多个数据库。

表 4-4：添加登录并授权数据库访问

任务	要求的角色	命令或过程	请参见
1. 为用户添加一个登录。	系统安全员	create login	第 15 页的 “创建登录帐户”
2. 将用户添加到一个或多个数据库。	系统管理员或数据库所有者	sp_adduser — 从数据库内部执行此过程。	第 61 页的 “向数据库添加用户”
3. 将用户添加到数据库中的组。	系统管理员或数据库所有者	sp_changegroup — 从数据库内部执行此过程。	• 第 66 页的 “更改用户组成员资格” • 《参考手册》中的 sp_changegroup
4. 为用户授予系统角色。	系统管理员或系统安全员	grant role	• 第 154 页的 “授予和撤消角色” • 《参考手册》中的 grant
5. 创建用户定义角色并将角色授予用户。	系统安全员	create role grant role	• 第 139 页的“为用户创建和指派角色” • 《参考手册》中的 grant • 《参考手册》中的 create role
6. 授予对数据库对象的访问权。	数据库对象所有者		第 6 章 “管理用户权限”

为远程连接建立 Kerberos 安全机制

Adaptive Server 将在连接到另一台服务器以执行远程过程调用 (RPC) 时以及通过组件集成服务 (CIS) 建立远程连接时充当客户端。

对于通过 Adaptive 服务器实现 RPC 执行的远程服务器登录，会在两个服务器之间建立一个物理连接。服务器使用该物理连接建立一个或多个逻辑连接，每个 RPC 对应一个逻辑连接。

Adaptive 服务器使用 Kerberos 第 5 版提供的凭据委托功能，对尝试通过 CIS 建立远程服务器连接的 Kerberos 登录支持端到端 Kerberos 鉴定。

凭据委托或票据转发能让 Kerberos 客户端在连接到服务器时委托凭据，从而允许服务器初始化 Kerberos 鉴定以便代表 Kerberos 客户端进一步连接到其它服务器。

连接到 Adaptive 服务器的 Kerberos 客户端可以请求对 Adaptive Server 进行远程过程调用 (RPC)，而且在对远程 Adapter Server 进行常规的分布式查询处理请求时，使用 Kerberos 凭据委托功能通过 CIS 实现。远程服务器登录不支持用于与远程 Adaptive 服务器之间的连接的 Kerberos 鉴定功能。有关配置 CIS Kerberos 鉴定的信息，请参见 《组件集成服务用户指南》的第 2 章 “了解组件集成服务”中的 “Configuration for Component Integration Services Remote Procedure Calls”（配置组件集成服务的远程过程调用）。

统一登录和远程服务器登录

如果本地服务器和远程服务器均设置为使用安全服务，则可以使用以下两种方法中的一种在这两台服务器上使用统一登录：

- 系统安全员在远程服务器上使用 sp_remoteoption 来将用户定义为 “trusted”。用户使用 “统一登录”获得对本地服务器的访问权，并执行远程服务器上的 RPC。此用户成为远程服务器的受托用户，不需提供口令。
- 当用户连接到本地服务器时，指定一个远程服务器口令。Open Client Client-Library/C 所附带的 ct_remote_pwd 例程提供了用于指定远程服务器口令的功能。请参见 Open Client Client-Library/C Reference Manual（《Open Client Client-Library/C 参考手册》）。

获取关于远程服务器的信息

sp_helpserver 可显示服务器的有关信息。如果不带参数运行 sp_helpserver，它将提供 sysservers 中列出的所有服务器的相关信息。可以指定某一特定服务器来接收有关该服务器的信息。语法为：

```
sp_helpserver [server]
```

例如，要显示有关 GATEWAY 服务器的信息，需执行：

```
sp_helpserver GATEWAY
```

连接到服务器并使用安全服务

isql 和 bcp 实用程序包括以下命令行选项，用于对该连接启用基于网络的安全服务：

- -R *remote_server_principal*
- -V *security_options*
- -Z *security_mechanism*

这些选项在以下段落中说明。

- -R *remote_server_principal* — 像为安全性机制定义那样指定服务器的主管名称。缺省情况下，服务器的主管名称与服务器的网络名（用 -S 选项或 DSQUERY 环境变量指定）相匹配。当服务器的主管名称和网络名不同时，必须使用 -R 选项。
- -V *security_options* — 指定基于网络的用户鉴定。使用此选项时，用户必须在运行实用程序之前登录到网络的安全系统。在这种情况下，如果用户指定了 -U 选项，用户必须提供安全性机制所知的网络用户名；使用 -P 选项提供的任何口令将被忽略。-V 后面可接 *security_options* 关键字母字符串选项，以启用其它安全服务。这些关键字母包括：
  - c — 启用数据保密性服务。
  - d — 请求凭据委托并转发客户端凭据。
  - i — 启用数据完整性服务。
  - m — 启用用于建立连接的相互鉴定。
  - o — 启用数据源加戳服务。
  - r — 启用数据重放检测。
  - q — 启用顺序混乱检测。
- -Z *security_mechanism* — 指定要用于连接的安全性机制名称。

在 *libtcl.cfg* 配置文件中定义安全性机制名称。如果不提供 *security_mechanism* 名称，则使用缺省机制。请参见所用平台的 Open Client/Server Configuration Guide （《Open Client/Server 配置指南》）。

如果使用 Client-Library 连接到 Adaptive Server，可以在连接到服务器之前定义安全性属性。例如，要检查消息序列，可设置 CS_SEC_DETECTSEQ 属性。有关通过 Client-Library 使用安全服务的信息，请参见《Open Client Client-Library/C 参考手册》。

使用用于客户端的安全性机制

Adaptive Server 在启动时，确定它支持的一套安全性机制。请参见第 101 页的 “确定所支持的安全服务和机制”。 Adaptive Server 必须从支持的安全性机制列表中选择要用于特定客户端的一种机制。

如果客户端指定一种安全性机制（例如，使用 isql 的 -Z 选项指定），则 Adaptive Server 会使用该安全性机制。否则，它使用 *libtcl.cfg* 文件中列出的第一个安全性机制。

获取有关可用安全服务的信息

Adaptive Server 允许您确定：

- Adaptive Server 支持哪些安全性机制和服务
- 当前会话的哪些安全服务处于活动状态
- 是否为会话启用了特定安全服务

确定所支持的安全服务和机制

系统表 syssecmechs 提供有关 Adaptive Server 所支持的安全性机制和安全服务的信息。当您查询时，将动态地建立该表。它包含以下这些列：

- sec_mech_name — 是安全性机制名称；例如，安全性机制可能是 “NT LANMANAGER”。
- available_service — 是安全性机制所支持的安全服务名称；例如，安全服务可能是 “统一登录”。

有关某一安全性机制的信息可能会占据表的若干行：每行列出该机制所支持的一种安全服务。

若要列出 Adaptive Server 支持的所有安全性机制及服务，请运行：

```
select * from syssecmechs
```

确定活动的安全服务

若要确定当前会话的哪些安全服务处于活动状态，请使用函数 show_sec_services：

```
select show_sec_services()

------------------------------------------------
        unifiedlogin mutualauth confidentiality
(1 row affected)
```

确定是否启用了某一安全服务

若要确定是否启用了某一特定的安全服务 （例如 “mutualauth”），请使用函数 is_sec_service_on，其中 *security_service_nm* 是可用的安全服务：

is_sec_service_on(*security_service_nm*)

使用在查询 syssecmechs 时返回的安全服务器。

例如，若要确定是否启用了 “mutualauth”，请执行：

```
select is_sec_service_on("mutualauth")

-----------
          1
(1 row affected)
```

结果为 1 表明已为会话启用了此安全服务。结果为 0 表明未使用此服务。

使用 Kerberos

Kerberos 是一种网络鉴定协议，它使用密钥密码技术，以便客户端可以通过网络连接向服务器证明其身份。当用户登录到操作系统或执行鉴定程序时，将获得用户认证。各个应用程序将使用这些认证执行鉴定。用户只需登录一次，而不必登录到每个应用程序。

Kerberos 假设密钥分发中心 (KDC) 正在运行并针对您的领域进行了适当的配置，而且客户端库安装在领域中的每个客户端主机下或主机上。有关配置信息，请参见文档和 Kerberos 软件附带的参考页。

Adaptive Server 通过以下库来支持 Kerberos：

- CyberSafe Kerberos 库
- MIT Kerberos 库 1.3.1 版
- 本机库

注释 要启用 Kerberos 的安全性选项，必须拥有 ASE_SECDIR，即 “安全和目录服务”软件包。

Kerberos 兼容性

表 4-5 显示哪个平台支持哪一种 Kerberos。

表 4-5：Adaptive Server Kerberos 互操作性

硬件平台	KDC 服务器	通用安全标准 (GSS) 客户端
Solaris 32	CSF、 AD 或 MIT	CSF、 MIT 或本机
Solaris 64	CSF、 AD 或 MIT	CSF、 MIT 或本机
Linux 32	CSF、 AD 或 MIT	MIT 或本机
Windows 32	CSF 或 AD	CSF
AIX 32	CSF	CSF

使用以下密钥读取互操作性矩阵：

- CSF — CyberSafe Ltd.
- AD — Microsoft Active Directory
- MIT — MIT 1.3.1 版

在 Kerberos 下启动 Adaptive Server

若要在 Kerberos 下启动 Adaptive Server，请将 Adaptive Server 名称添加到 KDC 中，并将服务键提取到键表文件。例如：

```
/krb5/bin/admin admin/ASE -k -t /krb5/v5srvtab -R"
addrn my_ase; mod
my_ase attr nopwchg; ext -n my_ase eytabfile.krb5"
Connecting as:admin/ASE
Connected to csfA5v01 in realm ASE.
Principal added.
Principal modified.
Key extracted.
Disconnected.
```

注释　也可以使用命令行上的口令鉴定管理员。在本例中，使用了 -k 选项，该选项指示管理员搜索 */krb5/v5srvtab* 文件 （使用 -t 选项指定）中是否有管理员和 Adaptive Server 键，而不是提示输入用于编写 shell 脚本的口令。

配置 Kerberos

不管使用哪一种 Kerberos，其配置过程都是相似的。

1 设置 Kerberos 第三方软件，并创建一个 Kerberos 管理员。为此，必须执行以下操作：

   a 在将要运行 Open Client Server 客户端或 Adaptive Server 的计算机上安装 Kerberos 客户端软件。以下客户端软件包已经过验证，可以运行：

   - CyberSafe TrustBroker 4.0
   - MIT Kerberos 1.3.1 版

   b 在一台单独的专用计算机上安装 Kerberos KDC 服务器。

   注释 CyberSafe TrustBroker 4.0、MIT Kerberos 1.3.1 版和 Microsoft Windows Active Directory 中的 KDC 已经通过验证，可与 Adaptive Server 一起使用。

   c 在 Kerberos 服务器上创建具有管理权限的管理员帐户。此帐号用于后续的客户端操作 （如从客户机创建主要帐号）。

   注释 在 Kerberos 客户端计算机上执行其余步骤。

2 为 Adaptive Server *ase120srv* 或 *ase120srv@MYREALM* 添加 Kerberos 主要帐户。

3 为主要帐户 *ase120srv@MYREALM* 提取 *keytab* 文件，并将其作为文件存储：

```
/krb5/v5srvtab
```

以下 UNIX 示例使用了可用于 CyberSafe 或 MIT Kerberos 的命令行工具 kadmin （此外还有 GUI 工具，可用于管理 Kerberos 及用户）：

```
CyberSafe Kadmin:
% kadmin aseadmin
Principal - aseadmin@MYREALM
Enter password:
Connected to csfA5v01 in realm ASE.
Command:add ase120srv
Enter password:
Re-enter password for verification:
Principal added.
Command:ext -n ase120srv
Service Key Table File Name (/krb5/v5srvtab):
```

```
Key extracted.
Command:quit
Disconnected.
```

在生产环境中，请控制对 *keytab* 文件的访问。如果用户可以读取 *keytab* 文件，则该用户就能创建一个服务器来模拟您的服务器。

使用 chmod 和 chgrp 命令，以使 */krb5/v5srvtab* 文件变为以下形式：

```
-rw-r----- 1 root sybase 45 Feb 27 15:42 /krb5/v5srvtab
```

在将 Active Directory 用作 KDC 时，请登录到域控制器以添加用户和 Adaptive Server 主管。可使用 “Active Directory 用户和计算机向导”来引导您完成创建用户及主管的过程。

提取用于 Adaptive Server 的 *keytab* 文件需要一个名为 ktpass 的可选工具，该工具包含在 Microsoft 支持工具软件包中。

对于 Active Directory，使用 ktpass 提取 *keytab* 是独立于创建主管操作的一个步骤。在 Windows 上，用于 Adaptive Server 的 *keytab* 文件与 CyberSafe 程序文件放在一起。例如，如果将 CyberSafe 软件安装在驱动器 C: 上，则 *c:\Program Files\CyberSafe\v5srvtab* 就是 Adaptive Server *keytab* 文件的预期位置。

4　为用户 “sybuser1”添加名为 “sybuser1@MYREALM”的 Kerberos 主管。

5　启动 Adaptive Server，并使用 isql 命令以 “sa”身份登录。以下步骤用于配置 Adaptive Server 参数，以便使用 Kerberos 安全服务，并创建用户登录帐户。以下步骤在 Windows 或 UNIX 计算机上都是相同的：

- 将配置参数 use security services 更改为 1：

  ```
  sp_configure 'use security services', 1
  ```

- 为用户 “sybuser1”添加新登录，然后添加用户：

  ```
  create login sybuser1 with password password
  ```

6　关闭 Adaptive Server，并修改管理文件和连接配置文件。

- 在 UNIX 平台上 — *interfaces* 文件位于 *$SYBASE/* 下，并包含一个类似于以下内容的条目：

  ```
  ase120srv
        master tli tcp myhost 2524
        query tli tcp myhost 2524
        secmech 1.3.6.1.4.1.897.4.6.6
  ```

在 Windows 平台上 — *sql.ini* 文件位于 *%SYBASE%\ini* 下，并包含一个类似于以下内容的等同的服务器条目：

```
[ase120srv]
master=TCP,myhost,2524
query=TCP,myhost,2524
secmech=1.3.6.1.4.1.897.4.6.6
```

- *libtcl.cfg* 或 *libtcl64.cfg* 文件位于 UNIX 平台上的 *$SYBASE/$SYBASE_OCS/config/* 中。“SECURITY”部分应该包含一个用于 CyberSafe Kerberos 客户端库的条目，与以下内容相似：

  ```
  [SECURITY]
  csfkrb5=libsybskrb.so secbase=@MYREALM
  libgss=/krb5/lib/libgss.so
  ```

  以下是 64 位 CyberSafe Kerberos 客户端库条目：

  ```
  [SECURITY]
  csfkrb5=libsybskrb64.so secbase=@MYREALM libgss= \
  /krb5/appsec-rt/lib/64/libgss.so
  ```

  对于使用 MIT Kerberos 客户端库的计算机，相应的条目与以下内容相似：

  ```
  [SECURITY]
  csfkrb5=libsybskrb.so
  secbase=@MYREALM
  libgss=/opt/mitkrb5/lib/libgssapi_krb5.so
  ```

  对于使用本机操作系统 （例如 Linux）提供的库的计算机，相应的条目与以下内容相似：

  ```
  [SECURITY]
  csfkrb5=libsybskrb.so secbase=@MYREALM
  libgss=/usr/kerberos/lib/libgssapi_krb5.so
  ```

  在 Windows 上 — *%SYBASE%\%SYBASE_OCS%\ini\libtcl.cfg* 文件包含如下条目：

  ```
  [SECURITY]
  csfkrb5=libskrb secbase=@MYREALM
  libgss=C:\WinNT\System32\gssapi32.dll
  ```

注释 `libgss=<gss shared object path>` 指定要使用的 GSS API 库。您必须直接定位到正在使用的 Kerberos Client 库，特别是当一台计算机上安装了多个版本时更是如此。

- 还要检查 *$SYBASE/$SYBASE_OCS/config/* 下的 *objectid.dat* 文件，并确保 *[secmech]* 部分包含用于 *csfkrb5* 文件的条目：

  ```
  [secmech]
  1.3.6.1.4.1.897.4.6.6 = csfkrb5
  ```

7 可以使用环境变量来替换 *keytab* 文件、 Kerberos 配置以及领域配置文件的缺省位置。这是 Kerberos 特有的行为，可能不会在所有平台上都能工作。

例如，在 CyberSafe UNIX 平台上使用 CSFC5KTNAME 环境变量来指定 *keytab* 文件：

```
% setenv CSFC5KTNAME /krb5/v5srvtab
```

对于 MIT Kerberos，等同的环境变量是 KRB5_KTNAME。

有关这些环境变量的信息，请参见供应商文档。

您可能需要为动态库搜索路径修改环境变量。在 UNIX 上，最常用的环境变量是 LD_LIBRARY_PATH；在 Windows 上， PATH 通常设置为包括 DLL 位置。可能需要修改这些环境变量，以使应用程序能够正确装载第三方对象。例如，在 C-shell 环境下，下面的命令将把 CyberSafe 32 位 *libgss.so* 共享对象的位置添加到搜索路径中：

```
% set path = ( /krb5/lib $path )
```

8 重新启动 Adaptive Server。您应会看到：

```
00:00000:00000:2001/07/25 11:43:09.91 server
Successfully initialized the security mechanism
'csfkrb5'.The SQL Server will support use of this
security mechanism.
```

9 以 UNIX 用户 “sybuser1” 的身份使用 isql （不带 -U 和 -P 参数）进行连接：

```
% $SYBASE/$SYBASE_OCS/bin/isql -Sase120srv -V
1>...
```

也可以使用加密选项：

```
$SYBASE/$SYBASE_OCS/bin/isql -Sase120srv –Vc
```

使用主管名

主管名是服务器向 Kerberos 密钥分发中心 (KDC) 进行鉴定时使用的名称。如果有多个正在运行的 Adaptive Server 实例，则必须为每个 Adaptive Server 使用不同的主管名。

指定 Adaptive Server 主管名

使用 DSLISTEN 和 DSQUERY 环境变量或 dataserver -s*server_name* 命令行选项来指定 Adaptive Server 名称。

使用 setenv 命令或 -k dataserver 选项来设置主管名。

缺省情况下，主要名称即 Adaptive Server 名称。若要指定不同的名称，请在启动 Adaptive Server 以使用 Kerberos 之前设置 SYBASE_PRINCIPAL：

```
setenv SYBASE_PRINCIPAL <name of principal>
```

一旦设置了 Adaptive Server 主管名， Adaptive Server 即使用此变量的值向 Kerberos 鉴定自身。

若要在启动 Adaptive Server 时指定 Adaptive Server 主管名，请使用：

```
-k <server principal name>
```

在启用了 Kerberos 安全性机制的情况下启动 Adaptive Server 时，Adaptive Server 首先会使用 -k 选项指定的主要名称进行 Kerberos 鉴定。如果未指定 -k 选项，则 Adaptive Server 在环境变量 SYBASE_PRINCIPAL 中查找主体名称。如果两者均未指定，则 Adaptive Server 使用服务器名进行鉴定。

如果 *keytab* 文件中存在主管名条目，则 Adaptive Server 接受使用不同服务器主管名的 Kerberos Open Client 连接。允许使用不同主管名的连接：

- 传递一个空字符串作为 -k 选项的参数，或者
- 将 SYBASE_PRINCIPAL 环境变量设置为 ""。例如：

  ```
  export SYBASE_PRINCIPAL=""
  ```

示例

在此示例中， Adaptive Server 名称是 “secure_ase”，领域名是 “MYREALM.COM”。 Adaptive Server 名称是使用 dataserver 的 -s 参数在命令行上指定的。当前领域是在 *libtcl.cfg* 中由 secbase 属性值指定的：

```
[SECURITY]
csfkrb5=libskrb.so libgss=/krb5/lib/libgss.so
secbase=@MYREALM.COM
```

缺省 Adaptive Server 主管名为 “secure_ase@MYREALM.COM”。如果 Adaptive Server *keytab* 文件中定义的主管名为 “aseprincipal@MYREALM.COM”，您可以使用下面的选项 1 或 2 设置服务器主管名，从而覆盖缺省 Adaptive Server 主管名：

- 选项 1，指定 -k "：

  ```
  %
  $SYBASE/$SYBASE_ASE/bin/dataserver -dmaster.dat
  -s secure_ase -k aseprincipal@MYREALM.COM
  ```

  通过 Kerberos 进行鉴定时使用的 Adaptive Server 主管名为“aseprincipal@MYREALM.COM”。

- 选项 2，设置 SYBASE_PRINCIPAL：

  ```
  setenv SYBASE_PRINCIPAL aseprincipal@MYREALM.COM
  $SYBASE/$SYBASE_ASE/bin/dataserver –dmaster.dat
  -s secure_ase
  ```

  通过 Kerberos 进行鉴定时使用的 Adaptive Server 主管名为“aseprincipal@MYREALM.COM”，即 *$SYBASE_PRINCIPAL* 的值。

- 选项 3，既不设置 -k 也不设置 SYBASE_PRINCIPAL：

  ```
  % $SYBASE/$SYBASE_ASE/bin/dataserver –dmaster.dat
  -s secure_ase
  ```

  通过 Kerberos 进行鉴定时使用的 Adaptive Server 主管名为“secure_ase@MYREALM.COM”。

使用 *sybmapname* 处理用户主要名称

sybmapname 将 Kerberos 环境中使用的外部用户主管名转换为 Adaptive Server 用户登录名的命名空间。您可以自定义 sybmapname 共享对象，并将 Kerberos 输入缓冲区中指定的名称映射为适合于登录到 Adaptive Server 输出缓冲区的名称。

使用 sybmapname 共享对象来执行用户主管名和 Adaptive Server 登录名之间的自定义映射。可以选择在服务器启动时装载此共享对象，在成功完成 Kerberos 鉴定之后、用户主要名称映射到 syslogins 表中的登录名之前，将调用此共享对象中包含的函数 syb__map_name。当要映射的用户主管名和登录名不同时，此函数很有用。

```
syb__map_name(NAMEMAPTYPE *protocol, char *orig,
int origlen, char *mapped, int *mappedlen)
```

其中：

- NAMEMAPTYPE *protocol — 是为使用此函数而保留的结构。
- char *orig — 是输入缓冲区，该缓冲区不以 null 终止。
- int origlen — 是输入缓冲区长度，应小于或等于 255 个字符。
- char *mapped — 是输出缓冲区，该缓冲区不应以 null 终止。

- int *mappedlen — 是输出缓冲区长度，应小于或等于 30。

如果映射成功， syb__map_name 将返回大于 0 的值，或者，如果未发生映射，则返回值 0，而当 syb__map_name 中发生错误时则返回小于 0 的值。发生错误时，将向 Adaptive Server 错误日志中写入报告映射失败的内容。

例如，若要鉴定 Adaptive Server 上的 Kerberos 用户，请执行下列操作：

1 配置 Adaptive Server 以使用 Kerberos 安全性机制。请参见第 102 页的 “使用 Kerberos”和 Open Client/Server 文档，以及 Sybase 网站 (http://www.sybase.com/detail?id=1029260) 上标题为 “为 Sybase 配置 Kerberos” (Configuring Kerberos for Sybase) 的白皮书。

   $SYBASE/$SYBASE_ASE/sample/server/sybmapname.c 中有一个示例文件 *sybmapname.c*。

2 修改 *sybmapname.c* 以执行您的逻辑。请参见第 112 页的 “使用 sybmapname 时的注意事项”。

3 使用提供的通用平台专用的 makefile 生成共享对象或 DLL。可能需要修改 makefile 以适合您的平台特定设置。

4 请将生成的共享对象放在 $LD_LIBRARY_PATH （UNIX 计算机）和 PATH 变量 （Windows 计算机）中指定的位置。“sybase”用户应该具有该文件的读取和执行权限。

注释 Sybase 建议只赋予 “sybase”用户读取和执行权限，并应当拒绝所有其它访问权限。

使用 Kerberos 鉴定来验证您的 Adaptive Server 登录

要使用 Kerberos 鉴定来检验您的 Adaptive Server 登录，请假设：

- *$SYBASE* 是指您的版本和安装目录。
- *$SYBASE_ASE* 指的是服务器二进制文件所在的 Adaptive Server 版本目录。
- *$SYBASE_OCS* 指的是 Open Client/Server 版本目录。

示例 1 如果客户端的主管名为 `user@REALM`，并且 syslogins 表中对应的条目为 `user_REALM`，则可以对 sybmapname 进行编码，以接受输入字符串 `user@realm`，并将输入字符串转换为输出字符串 `user_REALM`。

示例 2 如果客户端主体名称为 `user`，并且在 syslogins 表中的相应条目为 `USER`，则可以对 sybmapname 进行编码，使其接受输入字符串 `user` 并将此字符串转换为大写字符串 `USER`。

sybmapname 由 Adaptive Server 在运行时装载并使用其逻辑进行必要的映射。

以下操作和输出阐释了示例 2 中描述的 sybmapname 函数。应将包含 syb__map_name() 的自定义定义的 *sybmapname.c* 文件作为共享对象（或 DLL）来编译和生成，并最终放在合适的路径位置。在启用了 Kerberos 安全性机制的情况下启动 Adaptive Server。

对提供标识的加密文件 “票证授予票证 (TGT)”进行初始化：

```
$ /krb5/bin/kinit johnd@public
Password for johnd@public:
$
```

列出 TGT：

```
$ /krb5/bin/klist
   Cache Type:Kerberos V5 credentials cache
   Cache Name:/krb5/tmp/cc/krb5cc_9781
Default principal:johnd@public
```

以 “sa”身份登录并验证 “johnd”的用户登录名：

```
$ $SYBASE/$SYBASE_OCS/bin/isql -Usa -P
      -Ipwd`/interfaces
1>

1> sp_displaylogin johnd
2> go
No login with the specified name exists.
(return status = 1)

1> sp_displaylogin JOHND
2> go
Suid:4
Loginame:JOHND
Fullname:
Default Database:master
Default Language:
Auto Login Script:
Configured Authorization:
Locked:NO
Password expiration interval:0
Password expired:NO
Minimum password length:6
Maximum failed logins:0
Current failed login attempts:
Authenticate with:ANY
(return status = 0)
```

Kerberos 鉴定成功，使用 sybmapname 实用程序将小写的 johnd 映射到大写的 JOHND，并允许用户 johnd 登录到 Adaptive Server：

```
$ $SYBASE/$SYBASE_OCS/bin/isql -V -I'pwd'/interfaces
1>
```

使用 *sybmapname* 时的注意事项

对 sybmapname 进行编码时：

- 修改 *sybmapname.c* 样本程序时要小心。避免使用可能生成分段错误、可能调用 exit、可能调用 system calls、可能更改 UNIX 信号或进行任何阻塞调用的代码。不正确的编码或调用可能会干扰 Adaptive Server 引擎。

  注释 Sybase 不对 sybmapname 中的编码错误承担任何责任。

- 编码时应当小心，在不再引用指针之前请检查所有指针，并避免系统调用。您编写的函数必须为快速名称过滤函数。
- 不要使用 goto 语句，因为这些语句可能导致无法预料的副作用，具体取决于平台。
- 如果使用多个领域，在将用户主管名映射到合适的登录名以反映领域信息要小心。例如，如果您拥有的两个用户的用户主要名称分别为 userA@REALMONE 和 userB@REALMTWO，应将它们映射到登录名 userA_REALMONE 和 userB_REALMTWO，而不是 userA 或 userB。这样可区别两个属于不同领域的用户。

并发 Kerberos 鉴定

Adaptive Server 15.0.3 版支持并发 Kerberos 鉴定，而早期版本则在 Kerberos 鉴定过程中使用锁定机制来保护内部数据结构。

如果存在使用 Kerberos 鉴定的并发登录， Adaptive Server 现在将建立多个 Kerberos 鉴定会话。

15.0.3 版还解决了并发登录会话在 Kerberos 鉴定过程中可能会被阻塞的问题。Adaptive Server 以前的版本与 MIT 1.3.x 版和 1.4.x Kerberos GSSAPI 库一起使用时，将发生此并发问题。

为 LDAP 用户鉴定配置 Adaptive Server

LDAP 用户鉴定允许客户端应用程序将用户名和口令信息发送到 Adaptive Server，以便通过 LDAP 服务器 （而不是 syslogins）进行鉴定。通过 LDAP 服务器进行鉴定时，您可以使用全服务器范围的口令，而不使用 Adaptive Server 或特定于应用程序的口令。

如果希望对用户管理进行简化和集中，或者要避免用户管理不必要的复杂情况，则 LDAP 用户鉴定非常理想。

LDAP 用户鉴定可与符合 LDAP 协议标准版本 3 的目录服务器 （包括 Active Directory、 iPlanet 和 OpenLDAP Directory Server）一起使用。

为 LDAP 用户鉴定使用以下鉴定算法之一：

- 用于鉴定的组合型 DN，可用于 Adaptive Server 12.5.1 版或更高版本，或者
- 用于鉴定的搜索型 DN，可用于 Adaptive Server 12.5.2 版及更高版本。

这些算法在获取用户的区分名 (DN) 的方式上有所不同。

用于 LDAP 协议的主数据结构为 LDAP URL。

LDAP URL 会在 LDAP 服务器上指定一组对象或值。 Adaptive Server 使用 LDAP URL 来指定用于鉴定登录请求的 LDAP 服务器和搜索条件。

LDAP URL 使用以下语法：

```
ldapurl::=ldap://host:port/node/attributes [base | one | sub] filter
```

其中：

- *host* — 是 LDAP 服务器的主机名。
- *port* — 是 LDAP 服务器的端口号。
- *node* — 指定对象层次中搜索开始处的节点。
- *attributes* — 是要返回到结果集中的属性列表。每个 LDAP 服务器都可能支持不同的属性列表。
- base | one | sub — 限定搜索条件。base 指定搜索基准节点； one 指定搜索基准节点以及基准节点下面的一个子级别； sub 指定搜索基准节点及所有节点子级别。
- filter — 指定要鉴定的一个或多个属性。过滤器可以是简单的，如 `uid=*`，也可以是复合的，如 `(uid=*)(ou=group)`。

组合型 DN 算法

这是在使用组合型 DN 算法时的登录顺序：

1 Open Client 连接到 Adaptive Server 监听器端口。

2 Adaptive Server 监听器接受连接。

3 Open Client 发送一个内部登录记录。

4 Adaptive Server 读取该登录记录。

5 Adaptive Server 使用由主 URL 组成的 DN 和登录记录中的登录名绑定到 LDAP 服务器。此绑定还使用登录记录中的口令。

6 LDAP 服务器对用户进行鉴定，并返回成功或失败消息。

7 如果主 URL 指定一个搜索，那么 Adaptive Server 会向 LDAP 服务器发送搜索请求。

8 LDAP 服务器返回搜索的结果。

9 Adaptive Server 根据搜索结果接受或拒绝登录。

搜索型 DN 算法

这是在使用搜索型 DN 算法时的登录顺序：

1 Open Client 连接到 Adaptive Server 监听器端口。

2 Adaptive Server 监听器接受连接。

3 Open Client 发送一个内部登录记录。

4 Adaptive Server 读取该登录记录。

5 Adaptive Server 使用目录服务器访问帐户绑定到 LDAP 服务器。

在步骤 5 和 6 中建立的连接可能会在来自 Adaptive Server 的鉴定尝试之间持续存在，以便重复使用这些连接进行 DN 搜索。

6 LDAP 服务器对用户进行鉴定，并返回成功或失败消息。

7 Adaptive Server 根据登录记录中的登录名和 DN 查找 URL 向 LDAP 服务器发送搜索请求。

8 LDAP 服务器返回搜索的结果。

9 Adaptive Server 读取结果以从 DN 查找 URL 中获取属性值。

10 Adaptive Server 将属性值用作登录记录中的 DN 和口令，以绑定到 LDAP 服务器。

11　LDAP 服务器对用户进行鉴定，并返回成功或失败消息。

12　如果主 URL 指定一个搜索，那么 Adaptive Server 会向 LDAP 服务器发送搜索请求。

13　LDAP 服务器返回搜索的结果。

14　Adaptive Server 根据搜索结果接受或拒绝登录。

如果不满足这些搜索条件中的任意一个， Adaptive Server 都会向客户端报告常规登录失败。

如果不在主 URL 字符串或辅助 URL 字符串中指定搜索条件，就可以跳过步骤 12 和 13。鉴定完成，并显示步骤 11 返回的成功或失败消息。

配置 LDAP

可以在 Adaptive Server 中配置 LDAP 鉴定并将现有的 Adaptive Server 迁移到 LDAP。

❖ 在新安装的 Adaptive Server 中配置 LDAP

1　指定 Adaptive Server LDAP URL 搜索字符串和访问帐户值。

2　将 enable ldap user auth 设置为 2。

3　使用 LDAP 供应商提供的工具将用户添加到 LDAP 目录服务器中。

4　使用 create login 将用户添加到 Adaptive Server。也可以使用 sp_maplogin 自动创建用于鉴定的登录帐户或应用其它登录控制。

❖ 将现有 Adaptive Server 迁移到 LDAP

若要避免在现有服务器安装中出现服务中断，请将 Adaptive Server 迁移到 LDAP。

1　指定一个针对 Adaptive Server 的 LDAP URL 搜索字符串。

2　将配置参数 enable ldap user auth 设置为 1。

3　将用户添加到 LDAP 目录服务器中。

4　将所有用户添加到 LDAP 服务器时，将 enable ldap user auth 设置为 2，以要求通过 LDAP 执行所有鉴定，或者使用 sp_maplogin 将配置参数替换为登录控制。

LDAP 用户鉴定管理

使用 sp_ldapadmin 创建或列出 LDAP URL 搜索字符串，验证 LDAP URL 搜索字符串或登录，并指定访问帐户以及可调优 LDAP 用户鉴定 (LDAPUA) 相关参数。必须具有 SSO 角色才能执行 sp_ldapadmin。

请参见 《参考手册：命令》。

组合型 DN 示例

如果使用简单 LDAP 服务器拓扑和模式，则可以使用组合型 DN 算法进行用户鉴定。如果使用通过商业途径获得的模式（例如， iPlanet Directory Server 或 OpenLDAP Directory Server），则会将用户创建为 LDAP 服务器树上同一容器中的对象，并且 Adaptive Server 会根据对象的位置确定用户的 DN。但是， LDAP 服务器的模式有以下限制：

- 必须使用唯一地标识要鉴定用户的属性名指定过滤器。
- 必须将过滤器的属性指定为 `name=*`。星号为通配符。适合在过滤器中使用的属性名取决于 LDAP 服务器使用的模式。
- Adaptive Server 登录名与短用户名 （如 UNIX 用户名）相同。
- DN 使用短用户名，而不使用嵌入空格或标点符号的全名。例如，jqpublic 符合 DN 的限制，而 “John Q. Public”则不符合这一限制。

iPlanet 示例

LDAP 供应商所使用的对象名、模式和属性与以下示例中所用的对象名、模式和属性可能有所不同。可能存在许多 LDAP URL 搜索字符串，有效节点也可能会以本地方式扩展模式，或者以各不相同的方式使用它们：

- 此示例使用 `uid=*` 过滤器。为了组成 DN，Adaptive Server 会将通配符替换为要鉴定的 Adaptive Server 登录名，并在 LDAP URL 中将结果过滤器附加到节点参数中。得到的 DN 为：

  ```
  uid=myloginname,ou=People,dc=mycomany,dc=com
  ```

- 成功完成绑定操作后， Adaptive Server 会使用连接来搜索等效于登录名的属性名 （例如 `uid`）：

```
sp_ldapadmin set_primary_url,
'ldap://myhost:389/ou=People,dc=mycompany,dc=com??sub?uid=*'
```

- 此示例使用 OpenLDAP 2.0.25 中定义的模式，属性名为 `cn`。

  组成的 DN 为 `cn=myloginname,dc=mycompany,dc=com`:

  ```
  sp_ldapadmin set_primary_url,
  'ldap://myhost:389/dc=mycompany,dc=com??sub?cn=*'
  ```

搜索型 DN 示例

通过搜索型 DN，可以使用不符合组合型 DN 算法使用限制的 Active Directory 服务器或其它 LDAP 服务器环境。

- 使用 Windows 2000 Server 中通过商业途径获得的用户模式，对 Active Directory 服务器执行以下步骤。

  a　设置访问帐号信息：

```
sp_ldapadmin set_access_acct,
'cn=Admin Account, cn=Users, dc=mycompany, dc=com',
'Admin Account secret password'
```

  b　设置主 URL：

```
sp_ldapadmin set_primary_url, 'ldap://hostname:389/
```

  c　设置 DN 查找 URL 搜索字符串：

```
sp_ldapadmin set_dn_lookup_url,
'ldap://hostname:389/cn=Users,dc=mycompany,dc=com?distinguishedName
?one?samaccountname=*'
```

在 Windows 2000 上，短名称通常称为 “用户登录名”，并在缺省模式下被赋予属性名 samaccountname。这是用于匹配 Adaptive Server 登录名的属性名。用户的 DN 中包含一个带有标点符号和嵌入空格的全名（例如，`cn=John Q. Public, cn=Users, dc=mycomany, dc=com`）。Windows 上的 DN 不使用短名称，因此对于将 Active Directory 模式（缺省值）用作 LDAP 服务器的节点而言，搜索型 DN 算法较合适。主 URL 不指定搜索，而是依赖于绑定操作进行鉴定。

使用搜索过滤器限制 Adaptive Server 访问的示例

可以使用 LDAP URL 搜索字符串来限制对 LDAP 服务器上用户组的访问。例如，若要限制 accounting 组中用户的登录，可使用组合过滤器来限制对该用户组的访问，其中属性为 `group=accounting`。

- 以下 LDAP URL 字符串对 iPlanet 服务器使用组合型 DN 算法：

```
sp_ldapadmin set_primary_url,
'ldap://myhost:389/ou=People,dc=mycompany,
dc=com??sub?(&(uid=*)(group=accounting))'
```

  Adaptive Server 与 DN `uid=mylogin,ou=People,dc=mycompany,dc=com` 绑定在一起。成功绑定到此标识后，它会搜索以下内容：

```
"ou=People,dc=mycompany,dc=com??sub?(&(uid=mylogin)(group=accounting))"
```

  如果此搜索返回任何对象，则表明鉴定成功。

- 以下示例将 LDAP URL 字符串用于组合过滤器：

```
sp_ldapadmin set_primary_url,
'ldap://myhost:389/ou=people,dc=mycompany,dc=com??s
ub?(&(uid=*)(ou=accounting) (l=Santa Clara))'

sp_ldapadmin, set_primary_url,
'ldap://myhost:389/ou=people,dc=mycompany,dc=com??s
ub?(&(uid=*)(ou=Human%20Resources))'
```

LDAP 用户鉴定口令信息的更改

有两种与 LDAP 用户鉴定相关的信息性消息， Adaptive Server 从 LDAP 服务器获取这些消息并传递到客户端：

- 在使用 LDAP 鉴定机制登录到 Adaptive Server 时，如果 LDAP 用户鉴定口令即将到期，您将看到：

  ```
  Your password will expire in <number> days.
  ```

- 在 LDAP 服务器管理员重置了您的口令或者您的 LDAP 服务器口令已经到期之后，如果试图使用 LDAP 鉴定机制登录到 Adaptive Server，则会看到 4002 消息：

  ```
  Login failed
  ```

  如果启用了审计且打开了 errors 审计选项，则会将消息 4099 发送到审计日志：

  ```
  Your LDAP password has expired.
  ```

注释 请配置 LDAP 服务器以提供此附加信息。此外， Adaptive Server 必须支持将 LDAP 口令控制传送到 LDAP 客户端。

故障切换支持

如果由主 URL 指定的 LDAP 目录服务器出现严重故障，并且该服务器不再对网络请求做出响应，则 Adaptive Server 会尝试连接到由辅助 URL 指定的辅助 LDAP 目录服务器。Adaptive Server 使用 LDAP 函数 ldap_init 确定它能否打开与 LDAP 目录服务器的连接。 Null 或无效的主 URL 字符串会导致 Adaptive Server 尝试将故障切换到辅助 URL。LDAP 绑定或搜索操作返回的故障不会导致 Adaptive Server 将故障切换到辅助 URL。

Adaptive Server 登录和 LDAP 用户帐户

一旦启用了 LDAP 用户鉴定，选择并设置了鉴定算法和 URL 字符串，就必须配置用户帐户。LDAP 管理员在 LDAP 服务器中创建和维护帐户，数据库管理员在 Adaptive Server 中创建和维护帐户。或者，在将 Adaptive Server 与外部鉴定机制 （如 LDAP 服务器）集成时，数据库管理员也可以选择允许灵活使用登录帐户的管理选项。数据库管理员可以继续使用传统命令和过程，来管理 Adaptive Server 帐户角色、缺省数据库、缺省语言及其它特定于登录的属性。

表 4-6 介绍了登录时 Adaptive Server 对 syslogins 表进行的更新。这些更新假定已配置 LDAP 用户鉴定，未限制登录使用 LDAP，并且您未设置 create login 映射。

表 4-6：通过 LDAP 对 syslogins 进行的更新

syslogins 中是否存在行？	LDAP 服务器鉴定是否成功？	syslogins 中的更改
否	是	无更改，登录失败
否	否	无更改，登录失败
是	是	如果口令已更改，则更新行
是	否	无更改

辅助查找服务器支持

Adaptive Server 为 LDAP 服务器鉴定的 Adaptive Server 客户端提供不间断支持。您可以指定辅助 LDAP 查找服务器，当 LDAP 服务器发生故障或当计划的关机时间到来时可从主 LDAP 服务器故障切换到辅助服务器。

URL 的运行状况通过以下状态来监视：

- INITIAL （初始）— 表示未配置 LDAP 用户鉴定。
- RESET （重置）— 表示已通过 Adaptive Server 管理命令输入了 URL。
- READY （就绪）— 表示 URL 已准备好接受连接。
- ACTIVE （活动）— 表示 URL 已成功执行 LDAP 用户鉴定。
- FAILED （失败）— 表示连接 LDAP 服务器时出现问题。
- SUSPENDED （挂起）— 表示 URL 处于维护模式，并且不会再使用它。

以下事件序列描述了故障切换和手动故障恢复：

1 主 URL 组和辅助 URL 组都已配置并且处于 READY 状态。

2 已使用主服务器基础结构鉴定了连接。

3 主服务器故障，并且其状态变为 FAILED。

4 通过辅助服务器基础结构，连接自动开始鉴定。

5 LDAP 管理员修复了主服务器并使其恢复到联机状态。 Adaptive Server 管理员将主 LDAP 服务器状态更改为 READY。

6 已使用主服务器鉴定了新连接。

注释 一旦 Adaptive Server 进行故障切换并转移到辅助 LDAP 服务器，数据库管理员即必须手动激活主 LDAP 服务器，然后主服务器才能重新使用。

如果 Adaptive Server 在连接 LDAP 服务器时发生错误，则它会再尝试三次鉴定。如果错误仍然存在，则会将 LDAP 服务器标记为 FAILED。有关强制 Adaptive Server 进入重试循环的 LDAP 错误的信息，请参见 第 126 页的 “LDAP 用户鉴定错误的故障排除”。

使用 sp_ldapadmin 配置辅助查找 LDAP 服务器。

- 若要设置辅助 DN 查找 URL，请输入：

  ```
  sp_ldapadmin set_secondary_dn_lookup_url, <URL>
  ```

- 若要设置辅助 DN 查找 URL 的管理访问帐户，请输入：

```
sp_ldapadmin set_secondary_access_acct, <DN>, <password>
```

- 若要暂停使用主 URL 鉴定或辅助 URL 鉴定，请输入：

  ```
  sp_ldapadmin suspend, {primary | secondary}
  ```

- 若要激活主 URL 鉴定或辅助 URL 鉴定的设置，请输入：

  ```
  sp_ldapadmin activate, {primary | secondary}
  ```

- 要显示有关主/辅 LDAP 服务器设置和状态的详细信息，请输入：

  ```
  sp_ldapadmin list
  ```

  sp_ldapadmin list 结合了 list_access_acct 和 list_urls 先前的输出。它具有以下有关于主/辅服务器的所需输出：

  - 搜索 URL
  - 区分名查找 URL

- 访问帐户 DN
- 活动 [true | false]
- 状态 [ready | active | failed | suspended | reset]

Adaptive Server 12.5.4 版及更高版本包括以下支持辅助服务器的 sp_ldapadmin 选项。

- 若要显示辅助服务器的 DN 查找 URL，请输入：

  ```
  sp_ldapadmin list_urls
  ```

- 要显示辅助 DN 查找 URL 的管理访问帐户，请输入：

  ```
  sp_ldapadmin list_access_acct
  ```

- 若要显示子命令，请输入：

  ```
  sp_ldapadmin help
  ```

LDAP 服务器状态转换

表 4-7 — 表 4-12 列出了执行每个 sp_ldapadmin 命令时 LDAP 服务器的状态转换。

表 4-7 显示了执行 sp_ldapadmin set_URL 时的状态转换，其中 set_URL 表示以下命令之一：

- set_dn_lookup_url
- set_primary_url
- set_secondary_dn_lookup_url
- set_secondary_url

表 4-7：执行 sp_ldapadmin set_URL 时的状态转换

初始状态	最终状态
INITIAL	RESET
RESET	RESET
READY	READY
ACTIVE	RESET
FAILED	RESET
SUSPENDED	RESET

表 4-8 显示了执行 sp_ldapadmin suspend 时的状态转换。

表 4-8：执行 sp_ldapadmin suspend 时的状态转换

初始状态	最终状态
INITIAL	Error
RESET	SUSPENDED
READY	SUSPENDED
ACTIVE	SUSPENDED
FAILED	SUSPENDED
SUSPENDED	SUSPENDED

表 4-9 显示了执行 sp_ldapadmin activate 时的状态转换。

表 4-9：执行 sp_ldapadmin activate 时的状态转换

初始状态	最终状态
INITIAL	Error
RESET	READY
READY	READY
ACTIVE	ACTIVE
FAILED	READY
SUSPENDED	READY

以下表格显示了由 Adaptive Server 隐式执行的 LDAP 服务器状态转换。

表 4-10 显示了重新启动 Adaptive Server 时的状态转换：

表 4-10：重新启动 Adaptive Server 时的状态转换

初始状态	最终状态
INITIAL	INITIAL
RESET	RESET
READY	READY
ACTIVE	READY
FAILED	FAILED
SUSPENDED	SUSPENDED

如果 LDAP 服务器处于 READY 或 ACTIVE 状态，则 Adaptive Server 只尝试 LDAP 登录。表 4-11 显示了状态转换：

表 4-11：LDAP 登录成功时的状态转换

初始状态	最终状态
READY	ACTIVE
ACTIVE	ACTIVE

表 4-12 显示了 LDAP 登录失败时的状态转换：

表 4-12：LDAP 登录失败时的状态转换

初始状态	最终状态
READY	FAILED
ACTIVE	FAILED

LDAP 用户鉴定调优

可根据传入连接和 Adaptive Server-LDAP 服务器基础结构的负载对 Adaptive Server 选项进行配置和调优。可根据并发传入请求的数量来配置这些选项：

- 使用 sp_configure 设置 max native threads，它指定每个引擎的本机线程数。
- 使用 sp_ldapadmin 配置 max_ldapua_native_threads，它指定每个引擎的 LDAP 用户鉴定本机线程数。

根据网络和 Adaptive Server/LDAP 服务器基础结构的运行状况配置 set_timeout 选项 （该选项指示 LDAP 服务器绑定和搜索超时）。

配置 set_abandon_ldapua_when_full 选项，以指定当传入连接使用了 max_ldapua_native_threads 时的 Adaptive Server 行为：

使用这些 sp_ldapadmin 选项配置 LDAP 服务器以获得更好的性能：

- set_max_ldapua_desc — 管理 LDAPUA 连接请求的并发性。如果使用区分名算法，则将 set_max_ldapua_desc 设置为较大的数可加快 Adaptive Server 正在处理的 LDAPUA 连接。
- set_num_retries — 设置尝试次数。依据 Adaptive Server 和 LDAP 服务器之间的瞬时错误数调优此数值。可通过配置重试次数来取消瞬时错误。
- set_log_interval — 控制出于诊断目的发送到 Adaptive Server 错误日志的消息数。使用较低的数值会使错误日志杂乱，但有助于确定具体错误。如果使用较大的数值，则会减少发送到错误日志的消息数，但研究价值却不如前者。依据错误日志大小来调优 set_log_interval。

对登录映射增加更严格的控制

通过 sp_maplogin 将使用 LDAP 或 PAM 进行鉴定的用户映射到本地 Adaptive Server 登录。

注释 若要映射使用 Kerberos 进行鉴定的用户，请使用 sybmapname，而不是 sp_maplogin。

只有具有 sso_role 的用户才可以使用 sp_maplogin 创建或修改登录映射。

Adaptive Server 避免了登录的鉴定机制设置和使用该登录的映射之间的冲突。潜在的映射冲突由存储过程 sp_maplogin 或者命令 alter login 或 create login 进行检测。

这些控制不允许进行以下映射：

- 从一个 Adaptive Server 登录名映射到另一个登录名
- 从已经作为本地登录名存在的外部名称
- 映射到不存在的登录名

此外，如果鉴定机制是使用映射指定的，则该机制应与目标登录名中设定的鉴定机制相符。

如果目标登录的鉴定机制限制登录使用某个特定的鉴定机制，则使用映射指定的机制必须与为该登录指定的鉴定机制相匹配，或者与 “ANY” 鉴定机制相匹配。

当 sp_maplogin 检测到存在冲突时， sp_maplogin 将失败，并报告错误，指出所发生的冲突。

同样，alter login 和 create login 将检查是否存在某个可能与用户登录名的 authenticate with 选项存在冲突的映射。

当 alter login 或 create login 检测到冲突时，则会报告错误，指出与登录映射存在的任何冲突。

示例

示例 1 将 LDAP 用户映射到 Adaptive Server “sa” 登录。某公司采用了 LDAP 作为其所有用户帐户的存储库，且其安全性策略要求对所有用户进行 LDAP 鉴定，包括可能管理数百台 Adaptive Server 的数据库管理员 “adminA” 和 “adminB”。已启用审计，并且登录事件记录在审计追踪中。

若要将这些管理员帐户映射到 “sa”，请输入：

```
sp_maplogin LDAP, 'adminA', 'sa'
go
sp_maplogin LDAP, 'adminB', 'sa'
go
```

要求所有用户使用 LDAP 鉴定进行鉴定：

```
sp_configure 'enable ldap user auth', 2
go
```

当 “adminA” 在登录到 Adaptive Server 期间进行鉴定时，将在登录审计事件中记录与 “adminA” 关联的区分名，而不是仅记录 “sa”。这可以让每个单独执行的操作在审计追踪中标识出来。

由于 “adminA” 和 “adminB” 的口令是在 LDAP 服务器中设置的，因此无需在所有管理的 Adaptive Server 中都保留 “sa” 的口令。

此示例还允许在鉴定时使用不同的外部标识和口令，尽管它们在 Adaptive Server 中的操作仍然需要与 “sa” 帐户关联的特殊权限。

示例 2　结合使用 PAM 和 LDAP 将用户映射到应用程序登录。某公司同时采用了 PAM 和 LDAP 鉴定，但分别用于不同的目的。公司的安全策略将 LDAP 定义为一般用户帐户的鉴定机制，将 PAM 定义为特殊用户（例如中间层应用程序）的鉴定机制。中间层应用程序可以建立一个到 Adaptive Server 的连接池，以处理代表中间层应用程序用户的请求。

为 LDAP 和 PAM 用户鉴定配置 Adaptive Server：

```
sp_configure 'enable ldap user auth', 2
go
sp_configure 'enable pam user auth', 2
go
```

在本地建立一个 Adaptive Server 登录名 `appX`，并使其具有与中间层应用程序相适应的权限：

```
create login appX with password myPassword
go
alter login appX authenticate with PAM
go
```

这种鉴定机制并不采用在 “appX” 中硬编码一个简单口令然后使该口令在几个不同的 Adaptive Server 中保持一致的方法，而是开发了一个自定义 PAM 模块，该模块使用其它事实来鉴定中央存储库中的应用程序，以验证中间层应用程序的身份。

客户端应用程序登录 “appY” 需要使用用户的 LDAP 标识和口令对用户进行 LDAP 鉴定。使用 sp_maplogin 将所有经过 LDAP 鉴定的用户映射到登录 “appY”，

```
create login appY with password myPassword
go
sp_maplogin LDAP, NULL, 'appY'
go
```

“appY”的用户将通过其公司标识和口令接受鉴定，然后被映射到一个本地的 Adaptive Server 登录 “appY”，以执行数据库操作。鉴定是使用记录在审计追踪中的 LDAP 用户标识来进行的，数据库操作则是通过与应用程序登录 “appY”相适应的权限来执行的。

外部鉴定的登录映射

当您配置外部鉴定机制时，如果只有一个外部用户到内部 Adaptive Server 登录的映射，并且已成功对该映射进行鉴定，则 Adaptive Server 将更新内部登录的口令以匹配外部用户的口令。例如：

1 用户具有 Adaptive Server 登录名 user_ase （口令为 user_password）以及 LDAP 登录名 user_ldap （口令为 user_ldappasswd）。

这会生成 user_ldap 到 user_ase 的一对一映射。

2 当 user_ldap 使用 user_ldappassword 登录到 Adaptive Server 时，Adaptive Server 会将 user_ase 的口令更新为 user_ldappassword。

将 Adaptive Server 登录名映射到 LDAP 口令的好处是：如果 LDAP 服务器崩溃，用户可以用最近使用的 LDAP 口令进行登录。也就是说，当用户针对 Adaptive Server 鉴定具有用户名到 LDAP 口令的一对一映射时，用户就似乎具有对 Adaptive Server 的不间断鉴定，因为当口令用于鉴定登录时，它会在本地更新。

但是，当多个外部用户映射到本地用户时， Adaptive Server 不在本地更新口令。如果 LDAP 服务器崩溃，Adaptive Server 就无法对映射到单个 Adaptive Server 用户的多个外部用户进行鉴定。

LDAP 用户鉴定错误的故障排除

Adaptive Server 在与 LDAP 服务器通信时可能会发生以下暂时性的错误。通常，重新尝试连接可解决这些错误。如果重新尝试三次后错误仍然存在，则 Adaptive Server 将该 LDAP 服务器标记为 FAILED。

- LDAP_BUSY — 服务器繁忙。
- LDAP_CONNECT_ERROR — 连接过程中发生错误。
- LDAP_LOCAL_ERROR — 客户端发生错误。
- LDAP_NO_MEMORY — 无法在客户端分配内存。
- LDAP_OPERATIONS_ERROR — 服务器端发生错误。
- LDAP_OTHER — 未知错误代码。

- LDAP_ADMINLIMIT_EXCEEDED — 搜索超出限制。
- LDAP_UNAVAILABLE — 服务器无法处理请求。
- LDAP_UNWILLING_TO_PERFORM — 服务器将不会处理请求。
- LDAP_LOOP_DETECT — 在引用过程中检测到循环。
- LDAP_SERVER_DOWN — 无法访问服务器 （连接失败）。
- LDAP_TIMEOUT — LDAP API 因操作未在用户指定的时间内完成而失败。

瞬时错误和大量并发登录请求可能会导致错误日志中出现大量重复的错误消息。为提高日志的可读性，使用了以下错误消息记录算法：

1　如果是第一次记录某消息，则会记录它。

2　如果上次记录该消息的时间大于 3 分钟，则执行以下操作：

- 记录该错误消息。
- 记录自上次打印该消息以来该消息的重复次数。
- 以分钟为单位记录自输出该消息以来经过的时间。

因以下问题而导致的鉴定失败不视为 LDAP 错误，并且此类鉴定失败也不是重新尝试鉴定请求的条件：

- 由于口令错误或区分名无效而导致的绑定失败。
- 在成功绑定后搜索所返回的结果集为 0 或未返回任何属性值。

分析 URL 时发现的语法错误是在设置 LDAP URL 时捕获的，因此不能归于上述任何类别。

配置 LDAP 服务器

轻量目录访问协议 (LDAP) 的用户鉴定支持安全套接字层/传送层安全性 (SSL/TLS) 协议，从而能够在 Adaptive Server 和 LDAP 服务器之间安全地传输数据。

❖ **配置与 LDAP 服务器的连接**

1 确保所有受托根证书均位于同一文件中。

在您定义受托服务器之后， Adaptive Server 将配置安全连接，其中 *servername* 是当前 Adaptive Server 的名称。如果：

- 定义了 *$SYBASE_CERTDIR*， Adaptive Server 将从 *$SYBASE_CERTDIR/servername.txt* (UNIX) 或 *%SYBASE_CERTDIR%\servername.txt* (Windows) 装载证书。
- 未定义 *$SYBASE_CERTDIR*，则 Adaptive Server 将从 *$SYBASE/$SYBASE_ASE/certificates/servername.txt* (UNIX) 或 *%SYBASE%\%SYBASE_ASE%\certificates\servername.txt* (Windows) 中装载证书。

2 重新启动 Adaptive Server 以更改受托根证书文件。

3 使用 sp_ldapadmin（同时指定 *ldaps://* URL，而不是 *ldap://* URL）建立与 LDAP 服务器安全端口的安全连接。

4 通过简单的 TCP 连接建立 TLS 会话：

```
sp_ldapadmin 'starttls_on_primary', {true | false}
```

或者

```
sp_ldapadmin 'starttls_on_secondary', {true | false}
```

注释 LDAP 服务器连接没有 connect timeout 选项；如果 LDAP 服务器停止响应，则所有登录连接也将停止响应。

LDAP 用户鉴定改进

在 Adaptive Server 的早期版本中，如果需要修改证书颁发机构 (CA) 受托根文件，您必须重新启动 Adaptive Server 以使修改生效。 Adaptive Server 15.0.3 版及更高版本支持修改受托根文件，因此不必重新启动服务器。一个新的子命令 reinit_descriptors 可解除绑定 LDAP 服务器描述符，并重新初始化用户鉴定子系统。有关此选项的语法，请参见 《参考手册：过程》。

- 此命令需要系统安全员权限。
- 如果具有系统安全员权限的用户在未执行此命令的情况下修改了受托根文件，则管家实用程序杂事任务将使用新的杂事（设计为每隔 60 分钟重新初始化用户鉴定子系统一次）。

自动 LDAP 用户鉴定和故障恢复

Adaptive Server 15.0.3 提供了对辅助 LDAP 服务器的支持。在以前，在使出现故障的主 LDAP 服务器联机后，若要鉴定新的 LDAP 登录并将它们转移到主 LDAP 服务器，必须要手动激活 LDAP 服务器。

在 15.0.3 版及更高版本中， Adaptive Server 的管家实用程序中增加了一项新的杂事，用于自动激活 LDAP 服务器：'set_failback_interval' — 有关语法，请参见第 130 页的 “设置 LDAP 故障恢复时间间隔”。

sp_ldapadmin set_failback_interval 中的 set_failback_interval 选项设置激活出现故障的 LDAP 服务器的两次尝试之间的时间间隔；如果不设置此参数，则缺省值为 15 分钟。请参见 《参考手册：过程》中的 sp_ldapadmin。

如果主 URL 标记为 FAILED，则管家任务将尝试使用主访问帐户区分名 (DN) 和口令将其激活。如果尚未配置主访问帐户，则管家任务将尝试使用匿名绑定。如果绑定操作在第一次尝试时失败，则管家任务将按配置的重试次数重试绑定操作。如果绑定操作成功，则将主 URL 标记为 READY。

如果辅助 URL 标记为 FAILED，则管家任务将尝试按类似的方式激活辅助 URL。

sp_ldapadmin 中的 reinit_descriptors 选项在修改证书文件时执行，在这种情况下，它将每隔 60 分钟重新初始化 LDAP 用户鉴定子系统一次。

设置故障恢复间隔之后，管家任务将在每次执行其各项杂事时检查是否存在出现故障的 LDAP 服务器。如果找到出现故障的 LDAP 服务器，它将尝试在故障恢复时间间隔到期时激活 LDAP 服务器。

设置 LDAP 故障恢复时间间隔

sp_ldapadmin set_failback_interval 的语法如下，其中 *time_in_minutes* 为从 -1 到 1440 分钟 （24 小时）的值：

sp_ldapadmin 'set_failback_interval', *time_in_minutes*

- 值为 0 则表示手动进行故障恢复。也就是说，管家任务不会尝试自动对 LDAP 服务器进行故障恢复。您必须手动执行此任务。
- 如果值为 -1，则将故障切换时间间隔设置为 15 分钟 （缺省值）。
- 如果您不带任何参数发出 sp_ldapadmin 'set_failback_interval'，则 sp_ldapadmin 将显示为故障恢复时间间隔设置的值。
- 如果您不带任何参数发出 sp_ldapadmin，则 sp_ldapadmin 将在输出中包括故障恢复时间间隔：

```
 sp_ldapadmin
----------------
Primary:
    URL:                  ''
    DN Lookup URL:        ''
    Access Account:       ''
    Active:               'FALSE'
    Status:               'NOT SET
    StartTLS on Primary LDAP URL:'TRUE'
Secondary:
    URL:                  ''
    DN Lookup URL:        ''
    Access Account:       ''
    Active:               'FALSE'
    Status:               'NOT SET'
    StartTLS on Secondary LDAP URL:'FALSE'
Timeout value:            '-1'(10000) milliseconds
Log interval:             '3' minutes
Number of retries:        '3'
Maximum LDAPUA native threads per Engine:'49'
Maximum LDAPUA descriptors per Engine:'20'
Abandon LDAP user authentication when full:'false'
Failback interval:        '-1'(15) minutes
 (return status = 0)
```

示例

此示例将 LDAP 故障恢复时间间隔设置为 60 分钟：

```
sp_ldapadmin 'set_failback_interval' 60
```

此示例将 LDAP 故障恢复时间间隔设置为缺省值 15 分钟：

```
sp_ldapadmin 'set_failback_interval' -1
```

此示例显示为故障恢复时间间隔设置的值：

```
sp_ldapadmin 'set_failback_interval'
The LDAP property 'set_failback_interval' is set to '15
    minutes'.
```

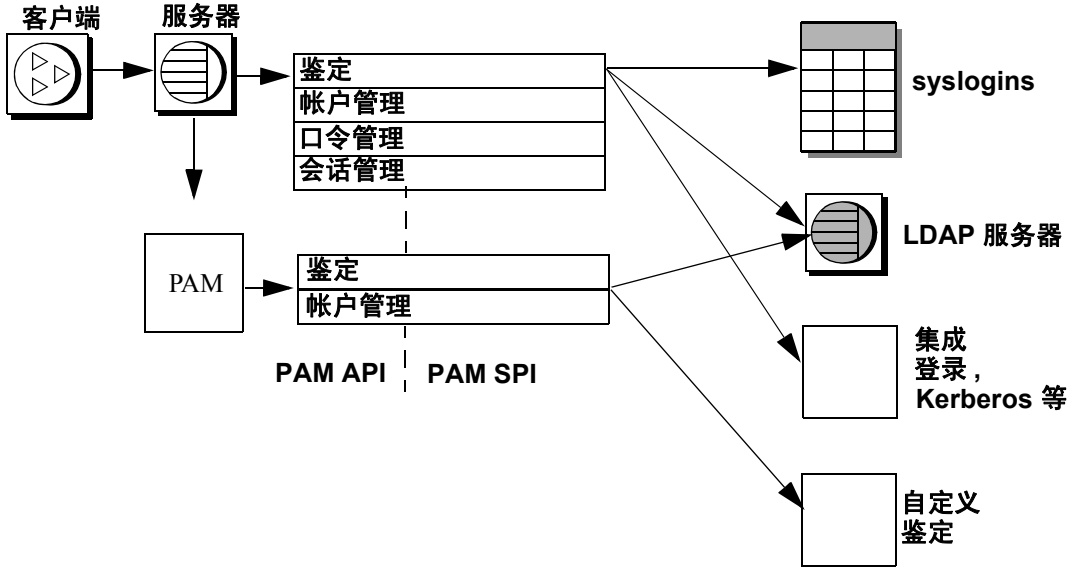
为使用 PAM 的鉴定配置 Adaptive Server

可插入鉴定模块 (PAM) 支持允许将多个鉴定服务模块叠加在一起，而无需修改需要鉴定的应用程序。

PAM 将 Adaptive Server 与 Solaris 和 Linux 操作系统集成在一起，并且简化了对用户帐户及鉴定机制的管理，从而降低了总拥有成本。用户可以自定义或编写他们自己的鉴定及授权模块。

注释　Linux 和 Solaris 平台上目前提供了 PAM 支持。有关 PAM 用户鉴定的详细信息，请参见操作系统文档。

图 4-2：PAM 体系结构



Adaptive Server 将登录名以及从登录包中获得的认证传递给 PAM API。PAM 装载在操作系统配置文件中指定的一个服务提供程序模块，并调用相应的函数来完成鉴定过程。

在 Adaptive Server 上启用 PAM

Linux 和 Solaris 中都包含预定义的 PAM 模块。可以使用这些模块之一，也可以创建自己的模块。在创建自己的模块时，请按照操作系统文档中有关创建 PAM 模块的指南进行操作。

注释 所创建的 PAM 模块应符合 RFC 86.0“Unified Login With Pluggable Authentication Modules (PAM)”。 Adaptive Server 支持 RFC 的鉴定管理模块。它不支持帐户管理模块、会话管理模块或口令管理模块。

配置操作系统

若要启用 PAM 支持，请按照下面的方法配置操作系统：

- 对于 Solaris，将下面的行添加到 */etc/pam.conf*：

```
ase auth required /user/lib/security/$ISA/pam_unix.so.1
```

- 对于 Linux，创建一个名为 */etc/pam.d/ase* 的新文件，并添加：

```
auth requried /lib/security/pam_unix.so
```

有关如何创建这些条目的详细信息，请参见操作系统文档。

在同一计算机上运行 32 位和 64 位服务器

$ISA 是一个允许 32 位和 64 位库同时运行的环境变量。

在 Solaris 32 位计算机上，用一个空字符串替换 $ISA；而在 64 位计算机上，则用字符串 “sparcv9”替换 $ISA。

若要同时运行 32 位和 64 位服务器，请将 32 位 PAM 模块放在一个目录中，然后将 64 位版本放到该目录的一个子目录中。

pam.conf 中的条目应该与以下内容类似：

```
$ ls /usr/lib/security/pam_sec.so.1
pam_sec.so.1 -> /SYBASE/pam_whatever_32bits.so.1

$ ls /usr/lib/security/sparcv9/pam_sec.so.1
pam_sec.so.1 -> /SYBASE/pam_sec_64bits.so.1

ase   auth   required
/usr/lib/security/$ISA/pam_sec.so.1
```

注释　$ISA 是 *pam.conf* 中允许的唯一变量。

为 PAM 用户鉴定配置 Adaptive Server

enable pam user auth 启用 PAM 用户鉴定支持：

```
sp_configure "enable pam user auth", 0 | 1 | 2
```

其中：

- 0 — 表示禁用 PAM 鉴定。这是缺省值。
- 1 — 表示 Adaptive Server 将首先尝试进行 PAM 鉴定，如果 PAM 鉴定失败，则使用 syslogins 鉴定。

- 2 — 表示只能使用 PAM 鉴定。

注释 启用 PAM 后，口令管理被委派给 PAM 服务提供程序。

Adaptive Server 登录和 PAM 用户帐户

在设置了 enable PAM user authentication 并完成了对 Adaptive Server 和操作系统的 PAM 配置之后，必须配置用户帐号。操作系统或网络安全管理员会在 PAM 服务提供程序中创建和维护用户帐户，数据库管理员会在 Adaptive Server 中创建和维护帐户。或者，在将 Adaptive Server 与外部鉴定机制 （如 PAM）集成时，数据库管理员也可以选择允许灵活使用登录帐户的管理选项。数据库管理员可以继续使用传统命令和过程，来管理 Adaptive Server 帐户角色、缺省数据库、缺省语言及其它特定于登录的属性。

表 4-13 介绍了登录时对 syslogins 进行的更新。该表假定\

已配置 PAM 用户鉴定，允许使用 PAM 进行登录，并且没有设置 create login 映射。

表 4-13：通过 PAM 对 syslogins 进行的更新

syslogins 中是否存在行？	PAM 鉴定是否成功？	syslogins 中的更改
否	是	无更改，登录失败
否	否	无更改，登录失败
是	是	如果口令已更改，则更新行
是	否	无更改

增强的登录控制

根据前面 LDAP 和 PAM 部分中讨论的方法，将 Adaptive Server 配置为允许使用全服务器范围的鉴定机制。还可以使用如下所述的 Adaptive Server 增强登录控制，将 Adaptive Server 配置为对该服务器上每个单独的登录分别指定鉴定机制。

当服务器在两种鉴定机制之间转换时，特定于登录的控制会很有用；对于本地服务器管理可能需要的特定于服务器的登录，特定于登录的控制也可能很有用：它们不与集中管理的用户登录关联。

强制鉴定

通过将以下参数用于 alter login 和 create login，可以强制登录使用特定的鉴定过程：

- ASE — 通过 syslogins 表中的口令使用 Adaptive Server 内部鉴定。
- LDAP — 通过 LDAP 服务器使用外部鉴定。
- PAM — 通过 PAM 使用外部鉴定。
- ANY — 缺省情况下，使用此鉴定方法对用户进行鉴定。使用 ANY 鉴定的用户意味着 Adaptive Server 将检查是否已经定义了外部鉴定机制，如果有，则使用该外部鉴定机制。否则，它将使用 Adaptive Server 的鉴定机制。

Adaptive Server 按照以下顺序检验外部鉴定机制：

1　LDAP。

2　可插入鉴定模块 (PAM)。如果同时启用了 LDAP 和 PAM，则决不会尝试对用户进行 PAM 鉴定。

3　如果 PAM 和 LDAP 都没有启用，则 Adaptive Server 使用 syslogins 对登录进行鉴定。

继续使用 syslogins 目录验证登录帐户 （例如 “sa”）。只有 SSO 角色可以设置对登录的鉴定。

例如，以下命令使用 alter login 鉴定登录：

```
alter login nightlyjob modify authenticate with ASE
sp_displaylogin "nightlyjob"
```

显示类似以下内容的输出：

```
Suid:1234
Loginname:nightlyjob
Fullname:Batch Login
Default Database:master
. . .
Date of Last Password Change:Oct 2 2003 7:38 PM
Password expiration interval:0
Password expired:N
Minimum password length:
Maximum failed logins:0
Current failed login attempts:
Authenticate with:ASE
```

使用 *sp_maplogin* 映射登录

使用 sp_maplogin 映射登录：

```
sp_maplogin (authentication_mech | null),
(client_username | null), (action | login_name | null)
```

其中：

- authentication_mech — 是为 sp_maplogin 中的 authenticate with 选项指定的有效值之一。
- *client_username* — 是一个外部用户名，它可以是操作系统名、LDAP 服务器的用户名，也可以是 PAM 库可识别的任何其它名称。空值表示任何登录名都有效。
- *action* — 表示 create login 或 drop。使用 create login 时，将在通过鉴定后立即创建登录。使用 drop 可删除登录。
- *login_name* 是 syslogins 中已经存在的 Adaptive Server 登录。

本示例将外部用户 “jsmith”映射为 Adaptive Server 用户 “guest”。一旦通过鉴定，“jsmith”就具有了 “guest”的特权。审计登录记录会同时显示 *client_username* 和 Adaptive Server 用户名：

```
sp_maplogin NULL, "jsmith", "guest"
```

如果尚未创建登录，则本示例将告知 Adaptive Server 为所有使用 LDAP 鉴定过的外部用户新建一个登录：

```
sp_maplogin LDAP, NULL, "create login"
```

显示映射信息

sp_helpmaplogin 显示映射信息：

```
sp_helpmaplogin [ (authentication_mech | null), (client_username | null) ]
```

其中：

- *client_username* — 是外部用户名。

如果未包括任何参数，则 sp_helpmaplogin 会显示有关当前登录到 Adaptive Server 的所有用户的登录信息。可以使用上面列出的参数，将输出限制到客户端用户名或鉴定机制的特定集合。

以下显示了所有登录的相关信息：

```
sp_helpmaplogin

authentication   client name   login name
--------------   -----------   ------------------
NULL             jsmith        guest
LDAP             NULL          create login
```

确定鉴定机制

使用 *@@authmech* 全局变量确定 Adaptive Server 使用的鉴定机制。

例如，如果允许 Adaptive Server 使用具有故障切换功能的 LDAP 用户鉴定 (enable ldap user auth = 2)，并且用户 “Joe” 为外部用户且鉴定机制设置为 ANY，则当 Joe 登录时， Adaptive Server 会尝试使用 LDAP 用户鉴定对 Joe 进行鉴定。如果 Joe 未能作为 LDAP 中的用户通过鉴定，则 Adaptive Server 会使用 Adaptive Server 鉴定对 Joe 进行鉴定；如果鉴定成功，则 Joe 可以成功登录。

@@authmech 全局变量值为：

```
select @@authmech
---------------------------------
ase
```

如果配置 Adaptive Server 以进行严格的 LDAP 用户鉴定 (enable ldap user auth = 2)，并将 Joe 作为有效用户添加到 LDAP 中，则当 Joe 登录时，@@authmech 的值为：

```
select @@authmech
---------------------------------
ldap
```

第 5 章

管理角色

本章包括有关在 Adaptive Server 中使用角色的信息。

为用户创建和指派角色

角色是让权限被授予者执行其工作的权限集合。 Adaptive Server 支持的角色使您可以强化个人责任。 Adaptive Server 提供了系统角色（如系统管理员和系统安全员）和由系统安全员创建并授予用户、登录配置文件或其他角色的用户定义角色。对象所有者可以为角色授予相应的数据库访问权限。

添加数据库用户的最后步骤是，为用户指派特定的角色并根据需要授予权限。有关权限的详细信息，请参见第 6 章“管理用户权限”。

系统定义角色

表 5-1 列出了系统角色、 grant role 或 revoke role 命令的 *role_granted* 选项使用的值以及具有此角色的用户通常执行的任务。

注释 以下各节详细描述了每个角色。

表 5-1：系统角色和相关任务

角色	*role_granted* 的值	说明
系统管理员	sa_role	管理和维护 Adaptive Server 数据库和磁盘存储
系统安全员	sso_role	执行与安全性相关的任务
操作员	oper_role	备份和装载全服务器范围内的数据库
Sybase 技术支持部门	sybase_ts_role	数据库结构的分析和修复

角色	*role_granted* 的值	说明
复制	replication_role	复制用户数据
分布式事务管理器	dtm_tm_role	跨服务器协调事务
高可用性	ha_role	管理和执行故障切换
监控和诊断	mon_role	管理和执行性能及诊断监控
Job Scheduler 管理	js_admin_role	管理 Job Scheduler
Job Scheduler 用户	js_user_role、 js_client_role	通过 Job Scheduler 创建和运行作业
实时消息传送	messaging_role	管理和执行实时消息传送
Web 服务	webservices_role	管理 Web 服务
密钥管理者	keycustodian_role	创建和管理加密密钥

注释 sa_role 可由具有 sa_role 的用户授予。所有其它系统角色都可由具有 sso_role 的用户授予。如果用户定义的角色已被授予了 sa_role 和其它系统角色，则该角色只能由既具有 sa_role 又具有 sso_role 的用户授予。

系统管理员特权

系统管理员：

- 处理非特定于应用程序的任务
- 在 Adaptive Server 的自由选择访问控制系统之外操作

通常将系统管理员角色授予个别 Adaptive Server 登录名。该用户执行的所有操作均可根据他/她各自的服务器用户 ID 进行跟踪。如果节点上的服务器管理任务由一个人执行，则可以改用随 Adaptive Server 一起安装的 “sa”帐户。安装时，“sa”帐户的用户可充当系统管理员、系统安全员或操作员角色。知道 “sa”口令的任何用户都可登录到此帐户并充当这些角色中的任何角色或所有角色。

让系统管理员在保护系统之外操作是一种安全预防措施。例如，如果数据库所有者意外删除了 sysusers 表中的所有条目，系统管理员可以恢复此表 （只要备份存在）。有几个命令只能由系统管理员发出。这些命令包括 disk init、 disk refit、 disk reinit、 shutdown、 kill、 disk mirror 、 mount、 unmount 及若干监控命令。

授予权限时，系统管理员被视作对象所有者。如果系统管理员授予操作另一用户的对象的权限，则相应的所有者名称将作为授权者在 sysprotects 和 sp_helprotect 输出中显示。

系统管理员在登录到数据库时自动充当数据库所有者的身份，并使用所有数据库所有者特权。无论为用户指派了任何别名，都会进行这种自动映射。系统管理员可执行通常为数据库所有者保留的任务，例如 dbcc 命令、诊断功能、读取数据页，以及恢复数据或索引。

系统安全员特权

系统安全员在 Adaptive Server 中执行与安全性相关的任务，包括：

- 授予系统安全员、操作员和密钥管理者角色
- 管理审计系统
- 更改口令
- 添加新登录名
- 删除登录名
- 锁定和解锁登录帐户
- 创建和授予用户定义的角色
- 管理基于网络的安全性
- 授予 set proxy 或 set session authorization 命令的使用权限
- 创建登录配置文件
- 管理加密

系统安全员可以访问任何数据库以启用审计，但通常情况下，系统安全员没有针对数据库对象的特殊权限（除了加密密钥和针对加密列的解密权限外。请参见《加密列用户指南》）。不过，sybsecurity 数据库除外，因为只有系统安全员才可以访问 sysaudits 表。还有几个只能由系统安全员执行的系统过程。

系统安全员可以修复用户对保护系统所进行的任何意外更改。例如，如果数据库所有者忘记了口令，系统安全员可以更改口令，让数据库所有者登录。

系统安全员与系统管理员一起负责登录管理。系统安全员负责管理登录名和登录配置文件。

系统安全员可以授予除 sa_role 外的所有系统角色。他们还可以创建用户定义角色并将相应角色授予用户、其它角色、登录配置文件或组。请参见第 139 页的 “为用户创建和指派角色”。

操作员特权

授予了操作员角色的用户可以在全服务器范围内备份和恢复数据库，而不必是每个数据库的所有者。操作员角色允许用户对任何数据库使用以下命令：

- dump database
- dump transaction
- load database
- load transaction
- checkpoint
- online database

Sybase 技术支持部门

使用技术支持角色，Sybase 技术支持部门的工程师就可以通过跟踪输出、一致性检查和修补数据结构，来显示内部内存结构和磁盘数据结构。此角色用于分析问题和手动恢复数据。解决这些问题所需的某些操作可能需要其它系统角色才能访问。 Sybase 建议只有在执行此类分析或修复时，系统安全员才将此角色授予知识渊博的 Sybase 工程师。

“复制”角色

维护 Replication Server 和 ASE Replicator 的用户需要 “复制”角色。有关此角色的信息，请参见 Replication Server Administration Guide（《Replication Server 管理指南》）和 ASE Replicator Users Guide（《ASE Replicator 用户指南》）。

分布式事务管理器角色

分布式事务管理器 (DTM) 事务协调器使用此角色允许系统存储过程跨服务器管理事务。使用 DTM XA 接口的客户端需要此角色。请参见 Using Adaptive Server Distributed Transaction Management Features （《使用 Adaptive Server 分布式事务管理功能》）。

高可用性角色

必须具有高可用性角色才能将高可用性子系统配置为通过命令和存储过程管理主服务器和协同服务器。请参见《在高可用性系统中使用 Sybase 故障切换》。

监控和诊断

管理 Adaptive Server 监控表需要此角色。必须具有此角色才能执行监控表远程过程调用和管理监控数据集合。请参见 Performance and Tuning Series: Monitoring Tables （《性能和调优系列：监控表》）。

Job Scheduler 角色

Job Scheduler 具有三种用于管理其操作权限的系统角色：

- js_admin_role — 使用此角色可管理 Job Scheduler、访问存储过程，以及修改、删除和执行 Job Scheduler 管理操作。
- js_user_role — 用户需要使用此角色通过 Job Scheduler 存储过程来创建、修改、删除和运行预定作业。
- js_client_role — 允许用户使用预定义的作业，但不允许创建或更改作业。

有关详细信息，请参见 《Job Scheduler 用户指南》。

实时消息传送角色

实时消息子系统 (RTMS) 使用此角色执行 msgsend、 msgrecv 和某些 sp_msgadmin 命令。有关详细信息，请参见 Messaging Services User's Guide （《消息传送服务用户指南》）。

Web 服务角色

Web 服务子系统使用此角色执行 create service、 create existing service、drop service 和 alter service 命令。请参见 《Web 服务用户指南》。

密钥管理者角色

密钥管理者角色负责密钥管理：创建和更改加密密钥、设置系统加密口令、为用户设置密钥副本等。请参见 《加密列用户指南》。

规划用户定义的角色

在实现用户定义角色之前，请确定：

- 希望创建的角色
- 每种角色的责任
- 每种角色在角色层次中的位置
- 层次中的哪些角色互斥，如果互斥，是在成员资格级还是在激活级互斥

遵循某种命名约定，以避免在创建用户定义角色时出现名称冲突。例如，可以在角色名中使用“_role”后缀。 Adaptive Server 不会检查此类限制。

直接向用户或登录配置文件授予的用户定义角色的名称不能与任何登录名或登录配置文件的名称相重复。如果角色必须与用户同名，为了避免冲突，可以创建一个新角色，并将原角色包含在新角色中，然后将这个新角色授予用户。

规划了要创建的角色及各角色之间的成员关系后，请确定如何根据用户的业务要求和责任来分配角色。

每个用户会话用户可激活的最大角色数为 127。

服务器范围内可创建的最大用户定义角色数是 992。

创建用户定义角色

具有 sso_role 的用户使用 create role 命令创建角色。请参见《参考手册：命令》中的 create role。

create role 命令只能在 master 数据库中使用。

如果使用了口令，则任何激活该角色的用户都必须指定口令。如果在登录期间将角色作为登录名的缺省角色激活或作为授予登录配置文件的自动激活的角色激活，则不能使用具有口令的角色。

例如，要不用口令创建 intern_role ，请输入：

```
create role intern_role
```

若要创建 doctor_role 并分配口令 “physician”，请输入：

```
create role doctor_role with passwd "physician"
```

只有系统安全员才能创建用户定义角色。

添加和删除角色口令

只有系统安全员才能添加或删除角色的口令。

使用 alter role 命令为系统角色或用户定义角色添加或删除口令：

```
alter role role_name
[add passwd password | drop passwd]
```

例如，若要为 oper_role 添加口令 “oper8x”，请输入：

```
alter role oper_role add passwd oper8x
```

若要删除角色口令，请输入：

```
alter role oper_role drop passwd
```

注释　当您向角色分配口令时，任何被授予该角色的用户都必须在激活角色时指定 Adaptive Server 的口令。

角色层次和互斥性

系统安全员可以定义角色层次，以便用户拥有某个角色时，也会拥有层次中较低级别的角色。当您将一个角色 role1 授予另一个角色 （例如 role2）时，将会设置层次结构，其中 role2 包含 role1。例如，“chief_financial_officer”角色可以包含 “financial_analyst”和 “salary_administrator”两个角色。

首席财务官可以执行所有任务，并查看工资管理员和财务分析员可以查看的所有数据。

此外，可以定义角色的互斥性，以强制实施职责的静态或动态分离策略。在以下情形时可以将角色定义为互斥：

- 成员资格 — 一个用户不能被授予两个不同的角色。例如，您也许不希望将 “payment_requestor” 和 “payment_approver” 这两个角色授予同一用户。
- 激活 — 一个用户不能激活（或启用）两个不同的角色。例如，用户可被授予 “senior_auditor” 和 “equipment_buyer” 角色，但不允许同时启用这两个角色。

系统角色和用户定义角色既可以按角色层次定义，也可以定义为互斥角色。例如，您可能希望有一个 “super_user” 角色包含系统管理员角色、操作员角色和技术支持部门角色。若要强制实施角色分离，您可能希望将系统管理员和系统安全员角色定义为在成员资格级互斥；即一个用户不能被同时授予这两个角色。

定义和更改角色互斥性

若要定义两个角色间的互斥性，请使用：

alter role *role1* { add | drop } exclusive { membership | activation } *role2*

例如，要定义 intern_role 和 specialist_role 在成员级互斥，输入：

```
alter role intern_role add exclusive membership
specialist_role
```

上面的示例限制在 intern_role 中具有成员资格的用户同时成为 specialist_role 的成员。

若要将 sso_role 和 sa_role 定义为在激活级别互斥，请输入以下命令，该命令禁止作为 sso_role 和 sa_role 成员的用户同时充当这两个角色：

```
alter role sso_role add exclusive activation sa_role
```

定义和更改角色层次

定义角色层次包括选择层次类型和角色，然后通过为其它角色授予角色来设计层次。

例如：

```
grant role intern_role to specialist_role
grant role doctor_role to specialist_role
```

这会为 “specialist” 授予 “doctor” 和 “intern” 的所有特权。

若要建立一个层次，层次中的“super_user”角色包括 sa_role 和 oper_role 系统角色，请指定：

```
grant role sa_role to super_user
grant role oper_role to super_user
```

注释　如果某个角色要求口令包含在其它角色中，则具有该角色（包含其它角色）的用户不必使用已包含角色的口令。例如，在上例中，假定“doctor 角色通常需要口令。具有“specialist”角色的用户不必输入“doctor”口令，因为“doctor”已包含在“specialist”中。只有最高级别的角色需要角色口令。

创建角色层次时：

- 不能将一个角色授予直接包含该角色的角色。这样可以防止重名。

  在上例中，由于“specialist”已经包含“doctor”，因此不能将“doctor”授予“specialist”。

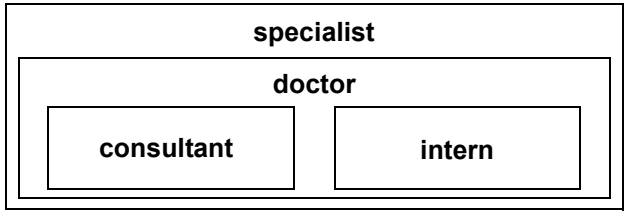
- 可以将一个角色授予不是直接包含该角色的另一个角色。

  例如，在图 5-1 中，即使“specialist”角色已经包含“doctor”角色，而“doctor”角色包含“intern”，也可以将“intern”角色授予“specialist”角色。如果随后从“specialist”中删除了“doctor”，则“specialist”仍包含“intern”。

  在图 5-1 中，“doctor”具有“consultant”角色权限，因为已将“consultant”授予“doctor”。“specialist”角色也具有“consultant”角色权限，因为“specialist”包含“doctor”角色，而“doctor”角色又包含“consultant”。

  但“intern”没有“consultant”角色的特权，这是因为“intern”不包含“consultant”角色（不论是直接还是间接的）。

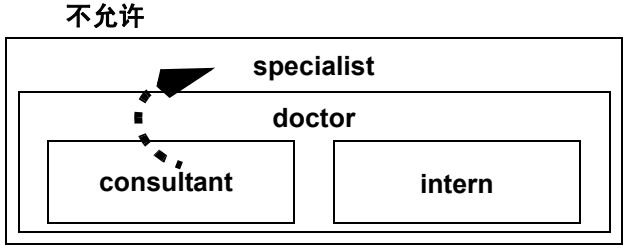
图 5-1：显式或隐式授予的特权



- 不能将一个角色授予该角色包含的另外一个角色。这可防止在角色层次中出现 “连环套”。

  例如，在图 5-2 中，不能将 “specialist”角色授予 “consultant”角色； “consultant”已包含在 “specialist”中。

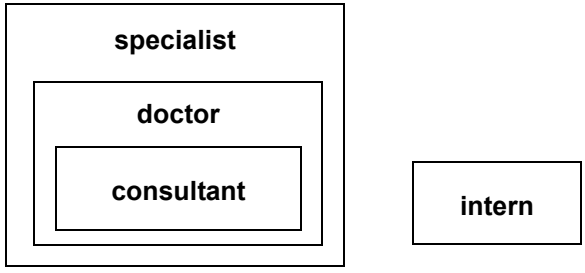
图 5-2：将角色授予授权者包含的角色



- 当系统安全员为用户授予一个包含其它角色的角色时，该用户将隐式获得该角色包含的所有角色的成员资格。但是，只有当用户在角色中拥有显式成员资格时，才能直接激活或停用该角色。
- 系统安全员不能将一个角色授予另一个在成员资格级显式或隐式与该角色互斥的角色。

  例如，在图 5-3 中，如果将 “intern”角色定义为在成员资格级与 “consultant”角色互斥，则系统安全员不可将 “intern”授予 “doctor”。

图 5-3：成员资格级的互斥性



- 用户只能直接激活或停用被授予的角色。

  例如，在图 5-3 所示的层次中，假定已经授予您 “specialist”角色。您拥有 “specialist”角色的所有权限，并且因为层次关系，您还隐式地拥有 “doctor”和 “consultant”角色的所有权限。然而，您只可激活 “specialist”角色。您不能激活 “doctor”或 “consultant”角色，这是因为并没有直接将它们授予您。请参见第 150 页的 “激活和停用角色”。

  从其它角色撤消角色与将角色授予其它角色类似。此操作删除了包含关系，且包含关系必须是直接包含。

例如：

- 如果系统安全员从“specialist”撤消“doctor”角色，则“specialist”不再包含 “consultant”角色或 “intern”角色。
- 系统安全员不能从 “specialist”撤消 “intern”角色，这是因为“intern”没有直接包含在 “specialist”中。

设置登录时的缺省激活角色

系统安全员可以使用 alter login 或 alter login profile 更改角色激活。

当用户登录到 Adaptive Server 时，其角色不一定处于活动状态，具体取决于将该角色设置为缺省角色的方式。如果该角色带有关联的口令，用户必须使用 set role 命令激活该角色。

系统安全员决定是否在登录时激活任何被缺省授予的角色，并使用 alter login profile 或 alter login 的 auto activated roles 属性为每个用户单独设置用户角色的缺省状态。各个角色只能更改自己的缺省设置。 auto activated roles 仅影响用户角色，不影响系统角色。

缺省情况下，授予的用户定义角色在登录时不激活，而授予的系统角色如果不带有与其关联的口令，则将自动激活。

以下示例显示如何在登录时自动激活角色 （如果它们不受口令保护）。

```
alter login mgr add auto activated roles
mgr_role, eng_role
```

以下示例显示如何使用登录配置文件在登录时自动激活角色 （如果它们不受口令保护）。 mgr_role 和 eng_role 必须授予 mgr_lp：

```
alter login profile mgr_lp add auto activated roles
mgr_role, eng_role
```

删除用户定义角色

作为系统安全员，使用以下命令删除角色：

drop role *role_name* [with override]

其中 *role_name* 是用户定义的角色的名称。

with override 撤消在服务器上的每个数据库中授予角色的所有访问特权。

如果不和 override 选项一起使用，您必须撤消在所有数据库中授予角色的所有特权，然后才能删除该角色。否则，删除命令将失败。若要撤消特权，请使用 revoke 命令。

删除角色前不必先删除成员资格。删除一个角色将自动删除具有此角色的任何用户的成员资格，无论是否使用了 with override 选项。

激活和停用角色

角色必须处于活动状态，才能具有访问特权（也就是说，不活动的角色没有特权）。缺省角色在登录时不能激活。具有口令的角色始终在登录时处于不活动状态。

激活或停用角色：

set role *role_name* [with passwd "*password*"] {on | off}

只有在要激活角色时才包括 with passwd 参数。请参见《参考手册：命令》。

例如，若要用口令 “sailing19” 激活 “financial_analyst” 角色，请输入：

```
set role financial_analyst with passwd "sailing19" on
```

仅在需要角色时才激活它们，不需要角色时使其失活。请记住，当 sa_role 激活时，用户将在任何所使用的数据库中承担数据库所有者的职责。

显示有关角色的信息

表 5-2 列出了要用于查找有关角色的信息的系统过程和函数。

表 5-2：查找有关角色的信息

显示有关信息	使用	请参见
角色名的角色 ID	role_id 系统函数	第 151 页的 “查找角色 ID 和名称”
角色 ID 的角色名	role_name 系统函数	第 151 页的 “查找角色 ID 和名称”
系统角色	show_role 系统函数	第 151 页的 “查看活动的系统角色”
角色层次和授予用户的角色	sp_displayroles 系统过程	第 152 页的 “显示角色层次”
角色在角色层次中是否包含其它角色	role_contain 系统函数	第 152 页的 “查看层次中的用户角色”
两个角色是否为互斥	mut_excl_roles 系统函数	第 152 页的 “确定互斥性”
为当前会话激活的角色	sp_activeroles 系统过程	第 152 页的 “确定角色激活”
是否已激活正确角色以执行过程	has_role 系统函数	第 153 页的 “检查存储过程中的角色”
登录 （包括授予的角色）	sp_displaylogin 系统过程	第 71 页的 “获取有关登录帐户的信息”
用户、组或角色权限	sp_helprotect 系统过程	第 187 页的 “权限报告”

查找角色 ID 和名称

若要在已知角色名时查找角色 ID，请使用：

role_id(*role_name*)

任何用户都可以执行 role_id。如果角色有效，则 role_id 返回角色在全服务器范围内的 ID (srid)。 syssrvroles 系统表包含一个含有角色 ID 的 srid 列和一个含有角色名的 name 列。如果角色无效，则 role_id 返回 NULL。

若要在已知角色 ID 时查找角色名，请使用：role_name:

role_name(*role_id*)

任何用户都可以执行 role_name。

查看活动的系统角色

使用 show_role 为指定登录显示当前活动的系统角色：

show_role()

如果没有激活任何系统角色，则 show_role 返回 NULL。如果用户是数据库所有者，并且在使用 setuser 模拟其他用户后执行 show_role，则 show_role 返回用户自己的活动系统角色，而不是所模拟用户的角色。

任何用户都可以执行 show_role。

注释 show_role 函数不提供有关用户定义角色的信息。

显示角色层次

您可以使用 sp_displayroles 查看授予您的登录名的所有角色，或查看以表格格式显示的整个角色层次树：

```
sp_displayroles {login_name | rolename [, expand_up | expand_down]}
```

任何用户都可以执行 sp_displayroles 来查看自己的角色。只有系统安全员可以查看有关授予其他用户的角色信息。

查看层次中的用户角色

可以使用 role_contain 确定任何指定角色是否包括其它指定角色：

```
role_contain (["role1", "role2"])
```

如果 *role2* 包含 *role1*，则 role_contain 返回 1。

任何用户都可以执行 role_contain。

确定互斥性

可以使用 mut_excl_roles 函数，确定为用户指派的任意两个角色是否互斥以及它们的互斥级别：

```
mut_excl_roles(role1, role2, {membership | activation})
```

任何用户都可以执行 mut_excl_roles。如果指定的角色或任何被指定角色包含的角色是互斥的，则 mut_excl_roles 返回 1；如果角色不互斥，则 mut_excl_roles 返回 0。

确定角色激活

若要查找 Adaptive Server 的当前登录会话的所有活动角色，请使用：

```
sp_activeroles [expand_down]
```

expand_down 显示任何授予用户的角色所包含的所有角色的层次。

任何用户都可以执行 sp_activeroles。

检查存储过程中的角色

可以在存储过程中使用 has_role 来保证仅具有特定角色的用户可以执行此过程。只有 has_role 可提供一种可靠方法，帮助防止对特定存储过程的不适当访问。

可以使用 grant execute，授予所有被授予指定角色的用户执行存储过程的权限。类似地，可使用 revoke execute 撤消这种权限。

然而， grant execute 权限不会阻止为没有指定角色的用户授予存储过程的执行权限。例如，若要确保绝不会为不是系统管理员的所有用户授予执行存储过程的权限，请在存储过程本身内使用 has_role，检查调用用户是否有执行该过程的正确角色。

has_role 为所需角色带一个字符串并在调用者拥有它时返回 1，否则返回 0。

例如，下面是一个使用 has_role 查看用户是否有 sa_role 角色的过程：

```
create proc test_proc
as
if (has_role("sa_role") = 0)
begin
    print "You don't have the right role"
    return -1
end
else
    print "You have System Administrator role"
    return 0
```

授予和撤消角色

定义了角色之后，可以将其授予服务器中的任何登录帐户或角色，只要不违反互斥性和层次规则即可。表 5-3 列出了与角色相关的任务、执行任务所要求的角色以及要使用的命令。

表 5-3：任务、要求的角色和所用命令

任务	要求的角色	命令
授予 sa_role 角色	系统管理员	grant role
授予 sso_role 角色	系统安全员	grant role
授予 oper_role 角色	系统安全员	grant role
授予用户定义角色	系统安全员	grant role
创建角色层次	系统安全员	grant role
修改角色层次	系统安全员	revoke role
撤消系统角色	系统安全员	revoke role
撤消用户定义角色	系统安全员	revoke role

授予角色

若要将角色授予用户或其他角色，请使用：

```
grant role role_granted [{, role_granted}...]
    to grantee [{, grantee}...]
```

其中：

- *role_granted* — 是所要授予的角色。可以指定任意多个要授予的角色。
- *grantee* — 是用户、角色或登录配置文件的名称。可以指定任意多个被授予者。

grant 语句中列出的所有角色都会授予所有被授予者。如果将一个角色授予另一角色，则会创建一个角色层次。

例如，若要为 Susan、 Mary 和 John 授予 “financial_analyst”和 “payroll_specialist”角色，请输入：

```
grant role financial_analyst, payroll_specialist
  to susan, mary, john
```

了解 *grant* 和角色

可使用 grant 命令，为所有授予了某个指定角色的用户授予对象权限，这与授予的角色是系统定义角色还是用户定义角色无关。这样，就可以限制具有以下任何一个角色的用户对对象的使用情况：

- 任何系统定义角色
- 任何用户定义角色

只能将角色授予一个登录帐户、另一个角色或一个登录配置文件。

对某个角色授予权限不会妨碍对*没有*指定角色的用户直接或通过组授予同样的权限。例如，若要确保只有系统管理员可以成功执行存储过程，请在该存储过程内使用 has_role 系统函数检查用户是否已被授予并激活了必需角色。请参见第 151 页的 “显示有关角色的信息”。

授予角色的权限将替换授予用户或组的权限。例如，假定已授予 John 系统安全员角色，并且已授予 sso_role 对 sales 表的权限。如果撤消 John 对 sales 的个人权限，他仍能够在 sso_role 处于活动状态时访问 sales，因为其角色权限替换了其个人权限。

撤消角色

使用 revoke role 撤消用户、其他角色和登录配置文件的角色：

revoke role *role_name* [{*, role_name*}...]from *grantee* [{*, grantee*}...]

其中：

- *role_name* — 是所要撤消的角色。可以指定任意多个要撤消的角色。
- *grantee* — 是用户名或角色名。可以指定任意多个被授予者。

revoke 语句中列出的所有角色都将从所有被授予者中撤消。

授予登录配置文件的角色

授予登录配置文件的角色可以由任何被分配了该配置文件的用户激活。请参见第 58 页的 “向登录配置文件授予角色”。

保护角色口令

在低于 15.7 版的 Adaptive Server 中，角色口令使用 Sybase 专有加密存储在 syssrvroles 系统表中。从 Adaptive Server 15.7 版开始，角色口令以 SHA-256 摘要的形式安全地存储在磁盘上。

当您将 Adaptive Server 升级到 15.7 或更高版本，并在升级后首次激活角色口令时，Adaptive Server 会加密角色口令并将其存储为 SHA-256 摘要。

不能对已经在 SHA-256 中加密的角色口令进行降级；如果降级，Adaptive Server 会截断角色口令并锁定角色。然后，管理员必须在降级后重置口令并解锁角色。

注释 在高可用性环境中，那些在主服务器上首次使用时升级的角色口令也会在其协同服务器上升级。

字符集考虑事项

在低于 15.7 的 Adaptive Server 版本中，口令在加密前使用服务器的缺省字符集。这个已经发生了变化，现在 Adaptive Server 会自动将口令转换为规范形式，即通用的标准化形式。这种自动转换可防止在您更改缺省字符集时由于字符映射不匹配而发生角色激活失败。

锁定的角色和 *syssrvroles*

可以使用 max failed_logins 选项将角色配置为在一定次数的角色激活尝试失败后自动锁定，或使用 alter role *rolename* lock 手动锁定。 Adaptive Server 在 syssrvroles 系统表中存储有关锁定角色的信息：

- lockdate — 指示角色何时被锁定。 lockdate 设置为锁定角色时的 datetime。
- locksuid — 指示谁锁定了角色。
- lockreason — 指示为何锁定角色。 lockreason 编码为一个整数，可以用国际化消息表示。每个原因都在 MSGDB 数据库中有一条添加的消息，以便用本地语言标识原因。

当角色解锁时， Adaptive Server 会将以下各项重置为 NULL。

值和说明为：

lockreason 的值	locksuid 的值	角色的 lockreason 的说明
NULL	NULL	角色未被锁定
1	alter role 的调用者的 suid	角色已由 suid 通过执行 alter role *rolename* lock 手动锁定
2	上次尝试角色激活导致角色被锁定的用户的 suid	角色已由 Adaptive Server 锁定，因为角色激活尝试的失败次数达到了最大失败登录次数

注释　如果您使用的是高可用性功能，则在更新 syssrvroles 列时主服务器和协同服务器都会升级。

对角色口令进行的登录口令策略检查

在 Adaptive Server 15.7 版中，适用于登录口令的口令复杂程度选项也适用于角色口令。以下选项检查哪些已扩展到角色口令：

- disallow simple passwords
- min digits in password
- min alpha in password
- min special char in password:
- min upper char in password
- min lower char in password
- systemwide password expiration
- password exp warn interval
- minimum password length
- maximum failed logins
- expire login

针对口令策略选项的高可用性支持

Adaptive Server 高可用性功能在主服务器和辅助服务器之间同步以下口令策略选项：

- disallow simple passwords
- min digits in password
- min alpha in password
- min special char in password:
- min upper char in password
- min lower char in password
- systemwide password expiration
- password exp warn interval
- minimum password length
- maximum failed login
- expire login
- keypair regeneration period
- keypair error retry wait
- keypair error retry count

Adaptive Server 使用 “password policy”定额属性检查主服务器和辅助服务器上的值是否一致。如果这些值在两个服务器上都相同，则高可用性建议检查会成功，否则会失败。例如：

```
sp_companion "MONEY1", do_advisory, 'all'
go
Attribute Name   Attrib Type  Local Value  Remote Value  Advisory
--------------   -----------  -----------  ------------  -------
expire login     password po   1            0             2
maximum failed   password po   3            5             2
min alpha in pa  assword po   10           12             2
```

输出的 advisory 列的值设置为 2，表示除非两个协同服务器上的值相同，否则用户不能进行聚簇操作。

sp_companion do_advisory 的输出也指示两个服务器上特定口令策略检查的不一致。

针对角色设置 Adaptive Server

安装

在使用角色功能之前，先确保 Adaptive Server 在 master 数据库中以及添加到 syssrvroles 表的列的事务日志中有额外的磁盘空间。数据库管理员可以使用 alter database 命令添加额外的空间。

若要确定角色口令更改所需的每页角色密度以及日志空间，请使用 sp_help syssrvroles 和 sp_helpdb。然后，可以比较以下值：

- 特定数量的口令更改之前和之后的日志空间的值
- 对 syssrvroles 更新日期的特定数量的 set role with passwd 命令

升级 Adaptive Server

在升级过程中，Adaptive Server 会自动将 locksuid、lockreason 和 lockdate 添加到 syssrvroles 中。这些列是可空的，而且在升级后其缺省值为 NULL。只有在需要时才设置值。

降级 Adaptive Server

当将 Adaptive Server 降级到 15.5 版时， Adaptive Server 会截断并锁定角色口令。此外， Adaptive Server 不支持将 allow password downgrade 用于角色口令。

降级后，管理员应该重置角色口令并在再次使用角色帐户之前将其解锁。

在降级过程中， Adaptive Server 将会：

- 截断角色口令并锁定角色
- 删除 sysattributes 中类 35 下面的所有属性以及类 35 本身。
- 删除 syssrvroles 中的 locksuid、 lockreason 和 lockdate 列

在单用户模式中执行 sp_downgrade 时会发生降级口令的操作。以 “-m” 命令行选项开头的 dataserver 以单用户模式启动服务器并仅允许系统管理员登录。

在下面的示例中，执行 sp_downgrade 会导致 “doctor_role”角色的口令被锁定和截断。管理员可以将该输出重定向到文件中，以便这些角色的口令可以重置：

```
1> sp_downgrade 'downgrade','15.5',1
2> go
Downgrade from 15.7.0.0 to 15.5.0.0 (command:'downgrade')

Checking databases for downgrade readiness.

There are no errors which involve encrypted columns.

Executing downgrade step 2 [dbcc markprocs(@dbid)] for :
- Database:master (dbid:1)
sql comman is:dbcc markprocs(@dbid)
...

Executing downgrade step 26 [delete statistics syssrvroles(password) if exists (select 1
from syssrvroles where password is not
null) begin print "Truncating password and locking following role(s)" select name from
syssrvroles where password is not null update
syssrvroles set password = null, status = (status | @lockrole) where password is not null
end update syscolumns set length = 30
where id = object_id('syssrvroles') and name = 'password' update syssrvroles set locksuid
= null, lockreason = null, lockdate = null
where locksuid is not null or lockreason is not null or lockdate is not null delete
syscolumns where id = object_id('syssrvroles')
and name in ('locksuid', 'lockreason', 'lockdate')] for :
- Database:master (dbid:1)
sql comman is:delete statistics syssrvroles(password) if exists (select 1 from
syssrvroles where password is not null) begin print
"Truncating password and locking following role(s)" select name from syssrvroles where
password is not null update syssrvroles set
password = null, status = (status | @lockrole) where password is not null end update
syscolumns set length = 30 where id =
object_id('syssrvroles') and name = 'password' update syssrvroles set locksuid = null,
lockreason = null, lockdate = null where
locksuid is not null or lockreason is not null or lockdate is not null delete syscolumns
where id = object_id('syssrvroles') and
name in ('locksuid', 'lockreason', 'lockdate')

Truncating password and locking following role(s)
 name
 ------------------------------------------------------------
 doctor_role

Executing downgrade step 27 [delete sysattributes where class = 35 delete sysattributes
where class = 39 update syslogins set lpid =
null, crsuid = null where lpid is not null or crsuid is not null delete syscolumns where
id = object_id('syslogins') and name in
('lpid', 'crsuid') delete syslogins where (status & @lp_status) = @lp_status update
```

```
syslogins set status = status & ~(@exempt_lock)
where (status & @exempt_lock) = @exempt_lock] for :
- Database:master (dbid:1)
sql comman is:delete sysattributes where class = 35 delete sysattributes where class =
39 update syslogins set lpid = null, crsuid
= null where lpid is not null or crsuid is not null delete syscolumns where id =
object_id('syslogins') and name in ('lpid',
'crsuid') delete syslogins where (status & @lp_status) = @lp_status update syslogins set
status = status & ~(@exempt_lock) where
(status & @exempt_lock) = @exempt_lock

...

 (return status = 0)
```

错误日志中会显示额外的消息，以标识在 sp_downgrade 期间发生的步骤以及可能发生的系统错误，例如，在下面的降级过程错误日志输出示例中：

```
00:0006:00000:00006:2011/06/28 06:21:23.95 server  Preparing ASE downgrade from 15.7.0.0
to 15.5.0.0.
00:0006:00000:00006:2011/06/28 06:21:24.12 server  Starting downgrading ASE.
00:0006:00000:00006:2011/06/28 06:21:24.12 server  Downgrade :Marking stored procedures
to be recreated from text.
00:0006:00000:00006:2011/06/28 06:21:26.13 server  Downgrade :Removing full logging
modes from sysattributes.
00:0006:00000:00006:2011/06/28 06:21:26.13 server  Downgrade :Downgrading data-only
locked table rows.
00:0006:00000:00006:2011/06/28 06:21:26.13 server  Downgrade :Removing full logging
modes from sysattributes.
00:0006:00000:00006:2011/06/28 06:21:26.13 server  Downgrade :Removing column
sysoptions.number.
00:0006:00000:00006:2011/06/28 06:21:26.13 server  Downgrade :Removing srvprincipal
column from sysservers system table
00:0006:00000:00006:2011/06/28 06:21:26.14 server  Downgrade :Removing 'automatic master
key access' configuration parameter.
00:0006:00000:00006:2011/06/28 06:21:26.14 server  Downgrade :Removing DualControl
sysattribute rows
00:0006:00000:00006:2011/06/28 06:21:26.14 server  Downgrade :Downgrading sysattributes
system table.
00:0006:00000:00006:2011/06/28 06:21:26.16 server  Downgrade :Downgrading syscomments
system table.
00:0006:00000:00006:2011/06/28 06:21:26.19 server  Downgrade :Truncated role password,
locked role and removed columns locksuid, lockreason, lockdate from syssrvroles
00:0006:00000:00006:2011/06/28 06:21:26.21 server  Downgrade :Removing catalog changes
for RSA Keypair Regeneration Period and Login Profile
00:0006:00000:00006:2011/06/28 06:21:26.21 server  Downgrade :Turning on database
downgrade indicator.
00:0006:00000:00006:2011/06/28 06:21:26.21 server  Downgrade :Resetting database version
indicator.
00:0006:00000:00006:2011/06/28 06:21:26.21 server  ASE downgrade completed.
```

运行 sp_downgrade 之后，请关闭 15.0.2 服务器，以避免新的登录或者其它可能修改数据或系统目录的操作。

如果您重新启动 Adaptive Server 15.7 版：

- 成功执行 sp_downgrade 并关闭服务器后， Adaptive Server 会再次执行内部升级操作，对系统表的任何更改都会升级到 15.7 版。
- 在启动您返回到的早期 Adaptive Server 版本之前，必须再次执行 sp_downgrade。

可以启用锁定角色和截断口令。在下面的示例中， sp_displayroles 的输出显示降级过程锁定了 “doctor_role”并截断其口令：

```
select srid,status,name,password from syssrvroles
go
suid   status  name                password
------ ------- ------------------- -----
33     2       doctor_role         NULL
```

以下命令可解锁角色：

```
alter role doctor_role unlock
```

以下命令可为角色设置新口令：

```
alter role doctor_role add passwd "dProle1"
```

现在，运行 sp_displayroles 会显示角色已被解锁并具有口令：

```
select srid,status,name,
"vers"=substring(password,2,1) from syssrvroles
go
suid   status  name                vers
------ ------- ------------------- -----
33     0       doctor_role         0x05
```

第 6 章

管理用户权限

本章描述用户权限的使用和实施情况。

概述

自由选择访问控制 (DAC) 允许基于用户标识、组成员资格和活动角色限制对对象和命令的访问。因为具有某种访问权限的用户，例如对象所有者，可以选择是否将这种访问权限传递给其他用户，所以此类控制是 “自由选择”的。

Adaptive Server 的自由选择访问控制系统可识别以下用户类型：

- 拥有一个或多个系统定义角色的用户：系统管理员、系统安全员、操作员和其他角色
- 数据库所有者
- 数据库对象所有者
- 其他用户

系统管理员（具有 sa_role 的用户）在 DAC 系统之外操作，并且在任何时候对所有数据库对象（加密密钥除外）都具有访问权限（请参见《加密列用户指南》）。系统安全员始终可访问 sybsecurity 数据库中的审计追踪表，以跟踪系统管理员的访问。

如果您拥有 sa_role，并在 master 数据库中发出 grant 命令，则 all 也会授予 create database、 set tracing 和 connect 的权限。

数据库所有者不能自动接受其他用户所拥有对象的权限；但可以：

- 通过 setuser 命令来假定数据库中某个用户的身份，可以临时获得此用户的所有权限。
- 通过 setuser 命令来假定某个对象所有者的身份，并使用 grant 命令授予此对象的所有权限，可以永久获得此特定对象的权限。

有关利用另一用户的身份来获得某个数据库或对象的权限的详细信息，请参见第 177 页的 “获取另一用户的权限”。

对象所有者可以将这些对象的访问权限授予其他用户，也可以授予其他用户将访问权限再传递给其他用户。可以使用 grant 命令授予用户、组和角色各种权限，也可使用 revoke 命令撤消这些权限。使用 grant 和 revoke 授予用户权限以：

- 创建数据库
- 在数据库内创建对象
- 执行特定命令，如 dbcc 和 set proxy
- 访问指定的表、视图、存储过程、加密密钥和列

grant 和 revoke 还可以用于设置系统表的权限。

对于 “Public”的缺省权限，则不需要使用 grant 或 revoke 语句。

某些命令可由任何用户随时使用，而不需要任何权限。其它命令则只能由特定状态的用户使用，且不能移交给其他用户使用。

是否能为可授权和撤消的命令分配权限取决于各个用户的角色或状态（例如作为系统管理员、数据库所有者、系统安全员或数据库对象所有者）以及是否已授予用户一个角色，此角色有权将此权限授予其他用户。

还可以使用视图和存储过程作为安全性机制。请参见第 191 页的 “使用视图和存储过程作为安全性机制”。

创建数据库的权限

只有系统管理员才能授权使用 create database 命令。接受 create database 权限的用户还必须是有效的 master 数据库用户，因为所有数据库都是在使用 master 时被创建的。

在多数安装情况下，系统管理员保留对 create database 权限的独占，以便对数据库的放置和数据库设备空间分配进行集中控制。在这些情况下，系统管理员代表其他用户创建新的数据库，然后将所有权移交给相关用户。

若要为另一用户创建数据库：

1　在 master 数据库中发出 create database 命令。

2　使用 use 命令切换到新数据库。

3　执行 sp_changedbowner。

更改数据库所有权

使用 sp_changedbowner 更改数据库的所有权。通常，系统管理员先创建用户数据库，在完成某些初始工作后再将所有权授予另一用户。只有系统管理员才能执行 sp_changedbowner。

Sybase 建议您在用户尚未添加到数据库中，且用户还未开始在数据库中创建对象之时，移交所有权。新的所有者在 Adaptive Server 中必须已有一个登录名，但不能是该数据库的用户，也不能在该数据库中有别名。在可以更改数据库的所有权之前，可能需要先使用 sp_dropuser 或 sp_dropalias；在可以删除用户之前，可能需要删除某些对象。

在要更改所有权的数据库中发出 sp_changedbowner。语法为：

sp_changedbowner *loginame* [, true]

以下示例使“albert”作为当前数据库的所有者，并删除作为旧的“dbo”的用户的别名：

```
sp_changedbowner albert
```

请包括 true 参数以将别名及其权限移交给新的 “dbo”。

注释　您不能更改 master/model/tempdb 或 sybsystemprocs 数据库的所有权，并且不应更改任何其它系统数据库的所有权。

数据库所有者特权

数据库所有者和系统管理员是唯一能够向其他用户授予对象创建权限（create encryption key 和 create trigger 权限除外，这两个权限只能由系统安全员授予）的用户。数据库所有者具有在该数据库内执行任何操作的全部特权，必须使用 grant 命令将权限显式授予其他用户。

以下命令的使用权限自动授予数据库所有者，不能移交给其他用户：

- checkpoint
- dbcc
- alter database
- online database
- drop database
- dump database
- dump transaction
- grant （对象创建权限）
- load database
- load transaction
- revoke （对象创建权限）
- setuser

数据库所有者可授予或撤消执行以下操作的权限：

- 使用以下命令：create default、 create procedure、 create rule、 create table、 create view。

  如果数据库所有者拥有 sa_role 并处于 master 数据库中，则他们可授予使用 create database、 set tracing 和 connect 的权限。

- all — 如果您是数据库所有者，则 all 会授予除 create database、create trigger 和 create encryption key 之外的所有 create 命令的权限。
- 系统表的缺省权限
- 使用 dbcc 命令：checkalloc、 checkcatalog、 checkdb、 checkindex、 checkstorage、 checktable、 checkverify、 fix_text、 indexalloc、 reindex、 tablealloc、 textalloc、 tune

数据库对象所有者特权

创建数据库对象 （表、视图、加密密钥或存储过程）的用户拥有该对象，并自动授予所有对象访问权限。对于除对象所有者以外的用户 （包括数据库所有者在内），不向他们自动授予此对象的所有权限，除非对象所有者或对此对象有 grant 权限的用户对其明确授权。

举例如下，假设 Mary 是 pubs2 数据库的所有者，并且已授予 Joe 在此数据库中创建表的权限。现在， Joe 创建了 new_authors 表，则他就是此数据库对象的所有者。

最初，只有 Joe 具有 new_authors 的对象访问权限。 Joe 可以将此表的对象访问权限授予其他用户或撤消此权限。

以下对象更改权限缺省为表的所有者，并且不能移交给其他用户：

- alter table
- drop table
- create index

使用 grant 和 revoke 命令可授予特定用于对于特定数据库对象的 select、insert、update、 delete、 references、 decrypt、 truncate table、 update statistics、delete statistics 和 execute 权限，可使用 grant with grant option 命令将这些权限移交给其他用户。

对对象 （表、视图、索引、存储过程、规则、加密密钥、触发器或缺省值）执行 drop 命令的权限，在缺省情况下授予给对象所有者，且不能移交给其他用户。

其他数据库用户的特权

由对象所有者、数据库所有者、已通过 grant 选项授权的用户、系统管理员或系统安全员授予其他数据库用户权限或撤消其权限。这些用户由用户名、组名或关键字 public 指定。

所有用户都将在激活分配给他们的角色后继承对这些角色授予的权限。

系统过程的权限

在存储系统过程的 sybsystemprocs 数据库中设置系统过程的权限。

与安全性相关的系统过程只能由系统安全员运行。其它某些系统过程只能由系统管理员运行。

某些系统过程只能由数据库所有者运行。这些过程可以确保执行过程的用户是从中执行这些过程的数据库的所有者。

其它系统过程可由已授予权限的任何用户执行。用户在所有数据库中都必须具有执行系统过程的权限，或者在任一数据库中都不具有这一权限。

sybsystemprocs..sysusers 中未列出的用户在 sybsystemprocs 中被看作是“guest”，并被自动授予多个系统过程的权限。若要拒绝向用户授予系统过程权限，系统管理员必须将此用户添加到 sybsystemprocs..sysusers 中，并发出适用于该过程的 revoke 语句。用户数据库的所有者不能从自己的数据库内部直接控制系统过程的权限。

授予和撤消权限

以下类型的权限由 grant 和 revoke 控制：

- 对象访问权限
- 选择函数的权限
- 执行命令的权限(P)
- 执行 dbcc 命令的权限
- 执行某些 set 命令的权限
- 系统表的缺省权限

每个数据库都有自己独立的保护系统。在某个数据库中有权使用某一命令并不意味着也有权在其它数据库中使用同一命令。

对象访问权限

对象访问权限控制访问某些数据库对象的某些命令的使用。例如，必须明确授予用户在 authors 表上有使用 select 命令的权限。对象访问权限由对象所有者 （以及系统管理员或系统安全员）授予和撤消，他们可以将这些权限授予其他用户。

表 6-1 列出了对象访问权限的类型及其适用对象。

表 6-1：权限及其适用对象

权限	对象
select	表、视图和列
update	表、视图和列
insert	表和视图
delete	表和视图
references	表和列
execute	存储过程
truncate table	表
delete statistics	表
update statistics	表
decrypt	表、视图和列
select	加密密钥

references 权限指可在 alter table 或 create table 命令中指定的参照完整性约束。 decrypt 权限指对加密列进行解密所需的权限。加密密钥的 select 权限指在 create table、alter table 或 select into 命令中使用加密密钥对列进行加密所需的权限。其它权限参见 SQL 命令。对象访问权限缺省授予对象所有者，或授予系统管理员或系统安全员对于加密列的 decrypt 和对于加密密钥的 select 权限，并且可将对象访问权限授予其他用户。

如果有多个用户授予某个特定用户访问某对象的权限，则在所有授权用户撤消该访问权限之前，该用户的访问权限将一直予以保留。如果系统管理员撤消了访问权限，则即使其他用户已授予此用户访问权限，此用户仍然无权访问。

使用 grant 命令授予对象访问权限。请参见 《参考手册：命令》。

具体标识

在会话期间 Adaptive Server 通过登录名来标识用户。此标识适用于服务器中的所有数据库。用户创建对象后，服务器会将所有者的数据库用户 ID (*uid*) 和创建者的登录名与 sysobjects 表中的该对象相关联。这些信息将对象具体标识为属于此用户，从而允许服务器在隐式授予对象权限时识别。

如果 Adaptive Server 用户创建了一个表，然后创建了一个访问该表的过程，则任何授予了执行此过程的权限的用户，将不再需要直接访问此对象的权限。例如，通过向用户 “mary” 授予 proc1 的权限，她虽然对于表 table1 没有显式的选择权限，但可以查看该表中的 id 和 descr 列：

```
create table table1 (id     int,
                     amount money,
                     descr    varchar(100))

create procedure proc1 as select id, descr from table1

grant execute on proc1 to mary
```

但是某些情况下，只有当对象被具体标识，隐式权限才有用。其一是涉及别名和跨数据库对象访问。

SQL92 标准一致性的特殊要求

如果已使用 set 命令来启用 ansi_permissions ，则 update 和 delete 语句还需要其它权限。表 6-2 对所需权限进行了总结。

表 6-2：用于更新和删除的 ANSI 权限

	所需权限：*将 ansi_permissions 设置为 “关闭”*	**所需权限：*将 ansi_permissions 设置为 “打开”***
update	针对要设置值的列的 update 权限	针对要设置值的列的 update 权限 和 针对出现在 where 子句中的所有列的 select 权限 set 子句右侧所有列的 select 权限
delete	表的 delete 权限	要删除其中的行的表的 delete 权限 和 针对出现在 where 子句中的所有列的 select 权限

如果 ansi_permissions 设置为打开状态，并试图在不具有其它 select 权限的情况下进行 update 或 delete ，则此事务被回退，并且收到一条错误消息。如果发生这种情况，对象所有者必须授予所有相关列的 select 权限。

授予对象访问权限示例

以下语句授予 Mary 和 “sales”组在 titles 表中执行插入和删除操作的权限：

```
grant insert, delete
on titles
to mary, sales
```

以下语句授予 Harold 使用存储过程 makelist 的权限：

```
grant execute
on makelist
to harold
```

以下语句将执行自定义存储过程 sa_only_proc 的权限授予具有系统管理员角色的用户：

```
grant execute
on sa_only_proc
to sa_role
```

以下语句授予 Aubrey 在 authors 表中选择、更新和删除的权限，并向其授权将相同权限授予其他用户：

```
grant select, update, delete
on authors
to aubrey
with grant option
```

撤消对象访问权限示例

以下两个语句都撤消所有用户 （表所有者除外）对 titles 表中 price 列和 total_sales 列的更新权限：

```
revoke update
on titles (price, total_sales)
from public
```

此语句撤消 Clare 更新 authors 表的权限。如果她将此权限转授给其他用户，则此语句同时撤消所有这些用户的此权限：

```
revoke update
on authors
from clare
cascade
```

以下语句撤消操作员执行自定义存储过程 new_sproc 的权限：

```
revoke execute
on new_sproc
from oper_role
```

授予 dbcc 命令的权限

系统管理员可以将执行 dbcc 命令的权限授予在 Adaptive Server 中不具有系统管理员级特权的用户和角色。这种**自由选择访问控制**使管理员能够控制对数据库对象或特定数据库级和服务器级操作的访问。

有关完整的 dbcc 语法，请参见 《参考手册：命令》。

服务器范围的和特定于数据库的 *dbcc* 命令

dbcc 命令可以是以下两种情况之一：

- 特定于数据库 – 对特定目标数据库执行的 dbcc 命令 （例如，checkalloc、 checktable、 checkindex、 checkstorage、 checkdb、 checkcatalog、 checkverify、 fix_text、 indexalloc、 reindex、 tablealloc 和 textalloc）。尽管这些命令是特定于数据库的，但只有系统管理员才能够授予或撤销执行它们的权限。
- 服务器范围内 – tune 等在整个服务器范围内有效并且与任何特定数据库均无关联的 dbcc 命令。缺省情况下，这些命令会在整个服务器范围内授予，并且与任何数据库都没有关联。

系统管理员可允许用户在所有数据库中执行 dbcc 命令，方法是使这些用户成为这些数据库中的有效用户。但是，向角色授予执行 grant dbcc 命令的权限比向单独用户授予该权限更加方便，因为这样可以使用户以“guest”用户的身份使用数据库，而不需要手动将每个用户添加到数据库中。

从安全管理的角度来看，系统管理员可能更希望在服务器范围内授权执行特定于数据库的 dbcc 命令。例如，可以针对名为 storage_admin_role 的用户定义角色对所有数据库执行 grant dbcc checkstorage，从而无需针对每个数据库中的 storage_admin_role 执行 grant dbcc checkstorage。

以下命令在服务器范围内有效，而不是特定于数据库的命令：

- 服务器范围的 dbcc 命令，如 tune。
- 在服务器范围内授予的特定于数据库的 dbcc 命令，例如授予 storage_admin_role 的 grant dbcc checkstorage。

数据库中 *dbcc* 的被授予者和用户

grant dbcc 和 revoke dbcc 可以用于数据库中的用户。

由于在数据库中第一次对角色执行 grant 时，角色即作为用户自动添加到数据库中，因此，当角色被授予 dbcc 权限时，不会再有其它要求。登录名必须是在其中进行授权的数据库中的有效用户。有效用户包括“guest”。

对于服务器范围的 dbcc 命令，登录名必须是 master 中的有效用户，并且系统管理员必须在 master 中授予权限。

对于特定于数据库的 dbcc 命令，登录名应是目标数据库中的有效用户。

系统表的权限

如同任何其它表的权限一样，可以由数据库所有者来控制系统表的使用权限。创建数据库之后，某些系统表的 select 权限将授予 public，而某些系统表的 select 权限只有管理员才有。对于其它表，部分列的 select 权限对 public 有限制。

若要确定特定系统表的当前权限，请执行：

```
sp_helprotect system_table_name
```

例如，若要检查 master 数据库中的 syssrvroles 权限，请执行：

```
use master
go
sp_helprotect syssrvroles
go
```

缺省情况下，包括数据库所有者在内的任何一个用户都不能直接修改系统表，而是由 Adaptive Server 提供的 T-SQL 命令和系统过程来修改系统表。此种作法有助于保证完整性。

警告！ 虽然 Adaptive Server 提供了修改系统表的机制，Sybase 仍强烈建议不要进行修改。

授予系统表和存储过程缺省权限

grant 和 revoke 命令包括 default permissions 参数。 installmodel 或 installmaster 不会授予对任何系统表的缺省权限 （请参见下表）。相反，Adaptive Server 在建立新的数据库时分配对系统表的缺省权限。此命令的部分语法为：

```
grant default permissions on system tables
revoke default permissions on system tables
```

其中， default permissions on system tables 指定当您从任何数据库发出此命令时，都可以授予或撤消以下系统表的缺省权限：

sysalternates	sysjars	sysquerymatrics	systhresholds
sysattributes	syskeys	sysqueryplans	systypes
syscolumns	syslogs	sysreferences	sysusermessages
syscomments	sysobjects	sysroles	sysusers
sysconstraints	syspartitionkeys	syssegments	sysxtypes
sysdepends	syspartitions	sysslices	
sysgams	sysprocedures	sysstatistics	
sysindexes	sysprotects	systabstats	

缺省权限对所有系统表上的 public 应用 select，但以下情况例外：

- 撤消 public 对 syscolumns(encrkeyid) 的 select
- 撤消 public 对 syscolumns(encrkeydb) 的 select
- 将对 syscolumns 的 select 授予 sso_role
- 撤消 public 的 sysobjects(audflags) 权限
- 将 sysobjects 的权限授予 sso_role
- 撤消 public 对 sysencryptkeys 的所有列的 select
- 将对所有 sysencryptkeys 列的 select on 授予 sso_role

如果从 master 数据库运行此命令，则授予或撤消以下系统表的缺省权限：

syscharsets	syslanguages	sysmessages	sysservers
sysconfigures	syslisteners	sysmonitor	syssessions
syscurconfigs	syslocks	sysprocesses	syssrvroles
sysdatabases	syslogin	sysremotelogins	systimeranges
sysdevices	sysloginrole	sysresourcelimits	systransactions
sysengines	syslogshold	syssecmechs	sysusages

此命令还进行以下更改：

- 撤消 public 对 sysdatabases(audflags) 的 select
- 撤消 public 对 syscolumns(encrkeyid) 的 select
- 撤消 public 对 syscolumns(encrkeydb) 的 select
- 将对 syscolumns 的 select 授予 sso_role
- 撤消 public 对 sysdatabases(deftabaud) 的 select
- 撤消 public 对 sysdatabases(defvwaud) 的 select
- 撤消 public 对 sysdatabases(defpraud) 的 select
- 撤消 public 对 sysdatabases(audflags2) 的 select
- 将对 sysdatabases 的 select 授予 sso_role。
- 撤消 public 对 syslogins(password) 的 select
- 撤消 public 对 syslogins(audflags) 的 select
- 将对 syslogins 的 select 授予 sso_role
- 撤消 public 对 syslisteners(net_type) 的 select
- 撤消 public 对 syslisteners(address_info) 的 select
- 将对 syslisteners 的 select 授予 sso_role
- 撤消 public 对 syssrvroles(srid) 的 select
- 撤消 public 对 syssrvroles(name) 的 select
- 撤消 public 对 syssrvroles(password) 的 select
- 撤消 public 对 syssrvroles(pwdate) 的 select
- 撤消 public 对 syssrvroles(status) 的 select
- 撤消 public 对 syssrvroles(logincount) 的 select
- 将对 syssrvroles 的 select 授予 sso_role
- 撤消 public 对 sysloginroles(suid) 的 select
- 撤消 public 对 sysloginroles(srid) 的 select
- 撤消 public 对 sysloginroles(status) 的 select
- 撤消 sso_role 对 sysloginroles 的 select

组合 *grant* 和 *revoke* 语句

可将特定权限分配给特定用户，或者，如果要将大多数特权授予大多数用户，则可先将所有权限分配给所有用户，然后再撤消特定用户的特定权限，这样做更为简便。

例如，数据库所有者可以将对 titles 表的所有权限授予所有用户，即发出：

```
grant all
on titles
to public
```

数据库所有者随后可以发出一系列 revoke 语句，例如：

```
revoke update
on titles (price, advance)
from public
revoke delete
on titles
from mary, sales, john
```

grant 和 revoke 语句须区分先后顺序：发生冲突时，最近发出的语句将取代所有其它语句。

注释 在 SQL 规则下，必须先使用 grant 命令，然后再使用 revoke 命令，而不能在同一事务中使用这两个命令。因此，如果先授予对某些对象的“public”访问权限，然后又撤消其中一个用户的访问权限，则在短暂时间内，此用户仍可以访问这些对象。若要防止这种情况发生，请使用 create schema 命令将 grant 和 revoke 子句并入同一事务中。

了解权限顺序及层次

grant 和 revoke 语句与其发出的先后次序有关。例如，如果先为 Jose 所在的组授予对 titles 表的 select 权限，然后撤消 Jose 对 advance 列的 select 权限，则 Jose 可以选择 (select) 除 advance 列之外的所有列，而组中的所有其他用户仍然可以选择 (select) 所有列。

应用于某组或角色的 grant 或 revoke 语句可更改已指派给该组中任何成员或角色的所有相互冲突的权限。例如，如果 titles 表所有者已将不同权限授予 sales 组的各个成员，并希望标准化，则此所有者可以发出以下语句：

```
revoke all on titles from sales
grant select on titles(title, title_id, type,
        pub_id)
    to sales
```

同样，如果已向 public 发出 grant 或 revoke 语句，则所有用户以前发出的所有权限只要与新权限发生冲突都会更改。

同一 grant 和 revoke 语句，如以不同顺序发出，则会产生完全不同的情况。例如，下面一组语句将导致 public 组中成员 Jose 对 titles 不具有任何 select 权限：

```
grant select on titles(title_id, title) to jose
revoke select on titles from public
```

相反，如颠倒顺序发出同一组语句，结果只有 Jose 才具有对 title_id 和 title 列的 select 权限：

```
revoke select on titles from public
grant select on titles(title_id, title) to jose
```

如果在 grant 命令中使用 public 关键字，则也包括自己在内。如果对对象创建权限执行 revoke，则除非您是数据库所有者，否则 public 也包括您自己在内。如果 revoke 对象访问权限，而自己不是对象所有者，则 public 也包括自己在内。您也许希望不具有自己所有的表的使用权限，而只具有基于此表的视图的访问权限。为此目的，必须发出 grant 和 revoke 语句，明确设置自己的权限。可以使用 grant 语句重新设置此权限。

Grant dbcc 和 set proxy 发出针对 fipsflagger 的警告

如果启用了 set fipsflagger 选项，在发出 grant dbcc 和 set proxy 命令时它们会发出以下警告：

```
SQL statement on line number 1 contains Non-ANSI
text.The error is caused due to the use of DBCC.
```

获取另一用户的权限

Adaptive Server 提供两种方法来获取另一用户的标识和权限状态：

- 数据库所有者可以使用 setuser 命令来 “模拟” 当前数据库中另一用户的标识和权限状态。请参见第 178 页的 “使用 setuser”。
- **代理授权**允许一个用户在整个服务器范围内使用另一用户的标识。请参见第 178 页的 “使用代理授权”。

使用 setuser

数据库所有者可使用 setuser 进行以下操作：

- 访问另一用户所属对象
- 授予他人另一用户所属对象的权限
- 创建将由另一用户拥有的对象
- 由于某些其它原因，临时假定另一用户的 DAC 权限

尽管 setuser 命令可使数据库所有者自动获得另一用户的 DAC 权限，但此命令不会影响已经授予的角色。

setuser 权限缺省授予数据库所有者，并且不能移交。被模拟的用户必须是数据库的授权用户。 Adaptive Server 会检查被模拟用户的权限。

系统管理员可以使用 setuser 创建将由另一用户拥有的对象。但是，系统管理员在 DAC 权限系统之外操作，因此，他们不需要使用 setuser 来获取另一用户的权限。 setuser 命令会一直有效，直到发出另一 setuser 命令、当前数据库更改或用户退出登录时为止。

语法为：

setuser ["*user_name*"]

其中， *user_name* 是数据库中要被模拟的有效用户。

若要恢复原标识，请在不为 *user_name* 赋值的情况下使用 setuser。

此例说明数据库所有者如何授予 Joe 读取 Mary 的 authors 表的权限：

```
setuser "mary"
grant select on authors to joe
setuser     /*reestablishes original identity*/
```

使用代理授权

利用 Adaptive Server 的代理授权功能，系统安全员可以将另一用户的安全环境的使用权限授予所选登录名，且应用程序可采用受控方式代表不同用户执行任务。如果某个登录名具有代理授权的使用权限，此登录名就可以模拟 Adaptive Server 中的任何其它登录名。

警告！ 使用另一用户标识的权限功能极强，应将其局限于受托的管理员和应用程序。可以使用 grant set proxy ... restrict role 来限制用户在切换标识时无法获取的角色。

执行 set proxy 或 set session authorization 的用户利用被模拟用户的登录名和服务器用户 ID 进行操作。登录名存储在 master..syslogins 的 name 列中，服务器用户 ID 存储在 master..syslogins 的 suid 列中。这些值在整个服务器上的所有数据库中都处于活动状态。

注释　set proxy 和 set session authorization 在功能上完全相同，可以互换使用。唯一的区别是 set session authorization 与 ANSI-SQL92 兼容，而 set proxy 是 Transact-SQL 扩展。

使用 set proxy 限制角色

授予 set proxy...restrict role 来限制在切换标识时无法获取的角色。

set proxy 的语法为：

```
grant set proxy to user | role
    [restrict role role_list | all | system]
```

其中：

- *role_list* — 限制目标登录名的角色列表。被授者必须具有此列表上的所有角色，否则 set proxy 命令将失败。
- *all* — 确保被授者只能对与被授者具有相同角色 （或角色子集）的用户运行 set proxy。
- *system* — 确保被授予者与目标登录名具有相同的系统角色集。

例如，以下语句将 set proxy 授予用户 “joe”，但限制他将标识切换为具有 sa、sso 或 admin 角色的任何用户 （但是，如果他已经具有这些角色，则他可以对具有这些角色的任何用户执行 set proxy）：

```
grant set proxy to joe
restrict role sa_role, sso_role, admin_role
```

如果 “joe”尝试将其标识切换为某个具有 admin_role 的用户 （在本示例中，为 Our_admin_role）时，除非他已经具有 admin_role，否则此命令失败：

```
set proxy Our_admin_role
Msg 10368, Level 14, State 1:
Server 's', Line 2:Set session authorization permission
denied because the target login has a role that you do
not have and you have been restricted from using.
```

“joe”被授予 admin_role 并重试此命令后，此命令成功：

```
grant role admin_role to joe
set proxy Our_admin_role
```

有关 set proxy 命令的详细信息，请参见 《参考手册：命令》。

执行代理授权

执行 set proxy 或 set session authorization 时，遵循以下规则：

- 不得在事务内执行 set proxy 或 set session authorization。
- 不能为另一用户的代理使用锁定的登录。例如，如果 “joseph”是一个锁定的登录名，则不允许执行以下命令：

  ```
  set proxy "joseph"
  ```

- 可以在任何允许使用的数据库中执行 set proxy 或 set session authorization。但所指定的 *login_name* 必须是数据库中的有效用户，或者，数据库必须要有为其定义的 “guest”用户。
- 只允许一个级别；若要模拟多个用户，则在模拟操作之间必须返回到原标识。
- 如果在某个过程内执行 set proxy 或 set session authorization ，则原标识会在退出此过程时自动恢复。

如果已授予某一登录名使用 set proxy 或 set session authorization 的权限，则可以设置代理来模拟另一用户。以下是语法，其中 *login_name* 是 master..syslogins 中的有效登录名。

set proxy *login_name*

或者

set session authorization *login_name*

为登录名加上引号。

例如，若要将代理设置为 “mary”，请执行：

```
set proxy "mary"
```

设置代理后，检查服务器中的登录名和数据库中的用户名。例如，假定登录名为 “ralph”，并具有 set proxy 授权权限。现在打算以 “sallyn” 和 “rudolph” 身份在 pubs2 数据库中执行某些命令。“sallyn” 在此数据库中有一个有效的名称 (“sally”)，而 Ralph 和 Rudolph 没有。但 pubs2 已定义了一个 “guest” 用户。则可以执行：

```
set proxy "sallyn"
go
use pubs2
go
select suser_name(), user_name()
go
---------------------------- ------------------
sallyn                       sally
```

若要更改为 Rudolph，必须首先恢复自己的标识。为此目的，请执行：

```
set proxy "ralph"
select suser_name(), user_name()
go
---------------------------- -------------------
ralph                        guest
```

注意：Ralph 是此数据库中的 “guest”。

然后执行：

```
set proxy "rudolph"
go
select suser_name(), user_name()
go
---------------------------- -------------------
rudolph                      guest
```

Rudolph 也是此数据库中的 guest，因为 Rudolph 不是此数据库中的有效用户。

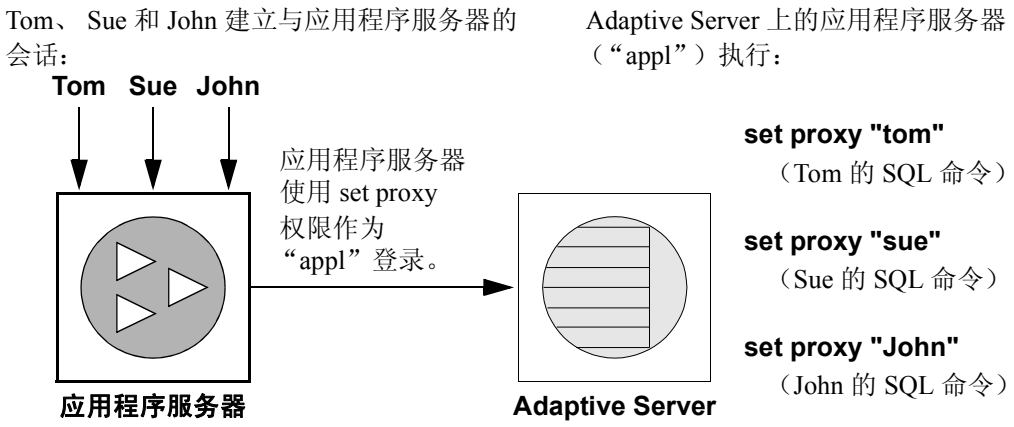
现在，模拟 “sa” 帐号。执行：

```
set proxy "ralph"
go
set proxy "sa"
go
select suser_name(), user_name()
go
------------------------- -------------------
sa                        dbo
```

应用程序的代理授权

图 6-1 显示使用通用登录名 “appl” 登录到 Adaptive Server 为多个用户执行过程和命令的应用程序服务器。一旦 “appl” 模拟 Tom，应用程序就有 Tom 的权限。同样，“appl” 分别模拟 Sue 和 John 时，应用程序便分别有 Sue 和 John 的权限。

图 6-1：应用程序和代理授权



更改数据库对象所有权

系统安全员或数据库所有者可以使用 alter... modify owner 命令来移交数据库对象的所有权。

此命令能让数据库管理员管理由于员工变动而带来的对象分配或分离数据库对象的创建所有权。例如，密钥管理者可以创建一个加密密钥，然后将该加密密钥的所有权移交给另一个用户。

支持的对象类型

以下对象的所有权可以从一个所有者移交给另一个所有者。未在下面列出的对象的所有权是无法更改的。

其所有权可以显式更改的对象：

- 用户表
- 代理表
- 视图
- 存储过程
- 用户定义的函数
- 缺省值
- 规则
- 用户定义的数据类型
- 加密密钥

其所有权无法显式更改的相关对象。以下对象将在所有权和显式移交的对象相同时被移交：

- 触发器

  如果 DBO 拥有的触发器是针对非 DBO 拥有的表/视图创建的，则该触发器的所有权是无法改变的。

- 在创建表/视图的过程中定义的声明性对象
  - 缺省值
  - 解密缺省值
  - 检查约束
  - 引用约束
  - 分区条件
  - 计算列

授权

- 系统安全员有权移交所有支持所有权移交的对象的所有权。
- 数据库所有者有权移交加密密钥以外的对象的所有权，但有以下限制：
  - 数据库对象所有者无法移交由数据库所有者具体拥有的对象的所有权。

    如果对象将数据库所有者的用户 ID 用作 sysobjects.uid，将空值或数据库所有者的用户名用作 sysobjects.loginame，则该对象会被标识为由数据库所有者具体拥有。
  - 别名为数据库所有者的用户不能移交由数据库所有者创建或由该用户具体拥有的对象的所有权。

    数据库所有者创建的对象在 sysobjects.loginame 中具有空值。有某个用户具体拥有的对象在 sysobjects.loginame 中使用该用户的用户名。

使用 sp_helpuser 可搜索并列出对象和相应的所有者。

移交所有权

所有权移交可以是特定于单个对象的，也可以在一个命令中移交多个对象。使用 preserve permissions 可保留对象的显式授予权限。

有关语法，请参见 《参考手册：命令》中的 alter...modify owner。

在下面的示例中，数据库所有者将 john 拥有的表移交给 eric。

```
alter table john.table_audit modify owner eric
```

若要将 john 拥有的所有表都移交给 eric，系统安全员可以执行：

```
alter table john.* modify owner eric
```

若要将 john 拥有的所有对象都移交给 eric，系统安全员可以执行：

```
alter all john.* modify owner eric
```

移交系统数据库中的对象的所有权

在更改以下由 Sybase 提供并管理的系统数据库中的对象的所有权时应小心：sybsecurity、 sybsystemdb、 model、 sybsystemprocs、 sybsyntax、 dbccdb 和 tempdb。不要更改由 Sybase 提供并管理的系统对象的所有权，如 （但不限于）带有 spt_ 前缀的用户表和带有 sp_ 前缀的系统存储过程。更改这些对象的所有权可能会使系统不可用。

移交数据库所有者对象的所有权

非系统对象的数据库所有者可以使用参数 dbo.object_name 移交所有权。不能使用 * 移交多个对象的所有权。

使用 *preserve permissions*

指定 preserve permissions 可保留对象的所有显式授予或撤消的权限。

例如， bill 向 mark 授予了表 `bill_table` 的 select 权限。然后， mark 向 john 授予了表 `bill_table` 的 select 权限。如果随后通过指定的 preserve permissions 将该表的所有权转交给 eric，则 mark 和 john 仍将具有对 `bill_table` 的权限。

在下面的示例中，系统安全员将视图 bill.vw_author 的所有权移交给 eric，同时保留所有已有的显式授予权限：

```
alter view bill.vw_author_in_ca modify owner eric
    preserve permissions
```

当指定 preserve permissions 时，不会保留隐式权限。

例如， bill 拥有具有加密列的表 bill.encr_table，而且 restricted decrypt permission 配置选项设置为 1。如果系统安全员向 bill 显式授予对 bill.encr_table 的 decrypt 权限， bill 将会拥有通过自己的所有权获得的权限 alter、delete、insert、reference、select 和 update。而且他还拥有通过系统安全员的显式授予获得的 decrypt 权限。系统安全员通过 preserve permissions 将 bill.encr_table 的所有权移交给 eric 后，bill 将失去对该表的所有权限，但 decrypt 权限除外。

如果不指定 preserve permissions，则在所有权移交后，以前的所有者将失去通过所有权隐式获得的对该对象的权限。通过给予对象的所有权，新的所有者将隐式获得权限。

注释　对于无法通过所有权获得的权限，如 decrypt 权限，系统安全员或数据库所有者必须再次向新的所有者显式授予这些对象的权限。

安全问题

系统安全员或数据库所有者应该注意可能发生的安全问题。

例如， alice 是 Accounting 数据库的用户，但无权访问 payroll 数据。她可以创建过程 alicep 用以从 Accounting.dbo.payroll 中选择姓名和工资，然后将对 alicep 的 execute 权限授予 public。如果系统安全员意外地将 alicep 的所有权更改为 bill（一个具有特权的用户，可通过 preserve permissions 选项访问 payroll 数据），则所有用户都可以通过执行恶意过程 alicep 访问 payroll 信息，因为所有权限都设置为在所有权更改后保留。

为避免非授权使用，系统安全员或数据库所有者可以使用 sp_helprotect 检查对某个对象的已有权限。

移交加密密钥的所有权

系统安全员和密钥所有者可以使用 alter encryption key 或 alter... modify owner 移交加密密钥。

有关 alter encryption key 命令的信息，请参见 《参考手册：命令》。

加密密钥副本所有者

使用 alter... modify owner 命令时，已被授予了密钥副本的用户不能是加密密钥的新所有者。

加密密钥的所有者发生更改后，密钥副本的被分配者不会更改。例如，用户 bill 拥有名为 bill.encrkey 的加密密钥，并创建该密钥的一个密钥副本用以授予 mark。在 bill 将 bill.encrkey 的所有权已经给 eric 后，mark 仍拥有 bill.encrkey 的副本。

权限报告

表 6-3 列出了用于报告有关代理、对象创建和对象访问权限信息的系统过程：

表 6-3：用于权限报告的系统过程

报告信息	使用
代理	系统表
用户和进程	sp_who
数据库对象或用户的权限	sp_helprotect
特定表的权限	sp_table_privileges
表中特定列的权限	sp_column_privileges

查询代理授权的 *sysprotects* 表

若要显示有关已授予 （或撤消）用户、组和角色的权限的信息，请查询 sysprotects 表。 action 列指定权限。例如， set proxy 或 set session authorization 的 action 值为 167。

可以执行以下查询：

```
select * from sysprotects where action = 167
```

查询结果为授予或撤消该权限用户的用户 ID （grantor 列）、有权限的用户的用户 ID （列 uid）和保护类型 （列 protecttype）。 protecttype 列可包含以下值：

- grant with grant 的值为 0
- grant 的值为 1
- revoke 的值为 2

有关 sysprotects 表的详细信息，请参见 《参考手册：构件块》。

显示有关用户和进程的信息

sp_who 显示有关当前所有 Adaptive Server 用户和进程的信息，或有关特定用户和进程的信息。sp_who 的结果包括 loginame 和 orignamе。如果用户正在某个代理下操作，则 orignamе 包含原始登录名的名称。例如，假设 “ralph” 执行以下命令，然后执行一些 SQL 命令：

```
set proxy susie
```

sp_who 为 loginame 返回 “susie”，并且为 orignamе 返回 “ralph”。

sp_who 查询 master..sysprocesses 系统表。在表包含服务器用户 ID 的列 (suid) 和原服务器用户 ID 的列 (origsuid)。

有关详细信息，请参见 《参考手册：过程》中的 sp_who。

报告数据库对象或用户的权限

使用 sp_helprotect 报告数据库对象或用户的权限，以及用户的指定对象权限 （可选）。任何用户都可以执行此过程。语法为：

```
sp_helprotect [name [, username [, "grant"
    [,"none"|"granted"|"enabled"|role_name]]]]
```

其中：

- *name* — 是表、视图或存储过程的名称，或者是当前数据库中用户、组或角色的名称。如果没有提供名称，则 sp_helprotect 报告数据库中的所有权限。
- *username* — 是当前数据库中的用户名。

  如果指定了 *username*，则只报告此用户对指定对象的权限。如果 *name* 不是对象，则 sp_helprotect 检查 *name* 是否是用户、组或角色。如果是用户、组或角色，则列出其权限。如果指定关键字 grant 且 *name* 不是对象，则 sp_helprotect 将显示通过 with grant option 授予的所有权限。
- grant — 显示使用 with grant option 授予 *name* 的权限。
- none — 忽略授予用户的角色。
- granted — 包括有关授予用户的所有角色的信息。
- enabled — 包括有关被用户激活的所有角色的信息。
- *role_name* — 只显示指定角色的权限信息，而不管此角色是否已授予此用户。

例如，假设已发出下面一系列 grant 和 revoke 语句：

```
grant select on titles to judy
grant update on titles to judy
revoke update on titles(contract) from judy
grant select on publishers to judy
    with grant option
```

若要确定 Judy 现在对 titles 表中各列的权限，请输入：

```
sp_helprotect titles, judy
```

```
grantor grantee type   action   object     column      grantable
------- ------ -----   ------   ------     ------      -------
dbo     judy   Grant   Select   titles     All         FALSE
dbo     judy   Grant   Update   titles     advance     FALSE
dbo     judy   Grant   Update   titles     notes       FALSE
dbo     judy   Grant   Update   titles     price       FALSE
dbo     judy   Grant   Update   titles     pub_id      FALSE
dbo     judy   Grant   Update   titles     pubdate     FALSE
dbo     judy   Grant   Update   titles     title       FALSE
dbo     judy   Grant   Update   titles     title_id    FALSE
dbo     judy   Grant   Update   titles     total_sales FALSE
dbo     judy    Grant   Update   titles     type        FALSE
```

第一行显示数据库所有者 （“dbo”）授予 Judy 选择 titles 表中所有列的权限。其余各行指示她只能更新显示中出现的列。Judy 不能将 select 或 update 权限授予任何其他用户。

若要查看 Judy 对 publishers 表的权限，请输入：

```
sp_helprotect publishers, judy
```

在此显示中， grantable 列显示为 TRUE，表示 Judy 可以将权限授予其他用户。

```
grantor grantee type   action   object     column      grantable
------- ------ -----   ------   ------     ------      -------
dbo     judy    Grant   Select  publishers   all         TRUE
```

报告特定表的权限

使用 sp_table_privileges 返回有关指定表的权限信息。语法为：

```
sp_table_privileges table_name [, table_owner
    [, table_qualifier]]
```

其中：

- *table_name* — 是表的名称，并且为必需。
- *table_owner* — 如果表所有者的名称不是 “dbo” 或执行 sp_table_privileges 的用户，则可用于指定表所有者的名称。
- *table_qualifier* — 是当前数据库的名称。

对要跳过的参数使用 null。

例如，此语句返回有关授予 titles 表的所有权限的信息：

```
sp_table_privileges titles
```

有关 sp_table_privileges 的输出的详细信息，请参见 《参考手册：过程》。

报告特定列的权限

使用 sp_column_privileges 可返回有关表中列的权限信息。语法为：

```
sp_column_privileges table_name [, table_owner
    [, table_qualifier [, column_name]]]
```

其中：

- *table_name* 是表名。
- *table_owner* — 如果表所有者的名称不是 “dbo” 或执行 sp_column_privileges 的用户，则可用于指定表所有者的名称。
- *table_qualifier* — 是当前数据库的名称。
- *column_name* — 是要查看权限信息的列的名称。

对要跳过的参数使用 null。

例如，此说明返回有关 publishers 表的 pub_id 列的信息：

```
sp_column_privileges publishers, null, null, pub_id
```

有关 sp_column_privileges 的输出的详细信息，请参见《参考手册：过程》。

使用视图和存储过程作为安全性机制

视图和存储过程可以作为安全性机制。可以通过视图或存储过程授予用户对数据库对象受控制的访问权限，而不授予他们对数据的直接访问权限。例如，可以向操作人员授予对更新 projects 表中成本信息的过程的 execute 权限，而不让此用户查看此表中的机密数据。若要使用此功能，必须拥有此过程或视图及其基础对象。如果没有基础对象，用户必须具有对象访问权限。有关何时需要权限的详细信息，请参见第 194 页的“了解所有权链”。

Adaptive Server 使用视图或过程时，根据需要进行权限检查。创建视图或过程时， Adaptive Server 不对基础对象进行权限检查。

使用视图作为安全性机制

通过视图，用户只能查询和修改所看到的数据。数据库的其余部分既不可见也不可访问。

必须明确授予或撤消访问视图的权限，而不管视图的基础表的权限。如果视图和基础表的权限由同一用户拥有，则不需授予基础表权限。授权访问视图而非基础表的用户无法看到视图中没有的基础表数据。

通过定义不同的视图并有选择地授予它们的权限，可以把用户或用户组合限制到不同的数据子集上。访问权限可以限制到：

- 基表中行的子集 （与值相关的子集）。例如，可以定义只包含商业与心理学书籍的视图，使某些用户无法看到其它类型书籍的信息。
- 基表中列的子集 （与值不相关的子集）。例如，可以定义包含 titles 表中所有行但不包含 price 列与 advance 列的视图，因为这些信息是敏感的。
- 基表的行列子集。
- 限定多个基表连接的行。例如，可以定义连接 titles、 authors 和 titleauthor 表的视图。此视图隐藏了作者的个人信息以及图书的价格信息。
- 基表中数据的统计信息。例如，可以只定义包含各类书籍平均价格的视图。
- 另一视图子集或视图和基表的某种组合的子集。

假设要阻止某些用户访问 titles 表中显示销售额和销售量的列。则可以创建 titles 表中不包括这些列的一个视图，然后将此视图的权限授予所有用户，而只将此表的访问权限授予销售部门：

```
grant all on bookview to public
grant all on titles to sales
```

如果不使用视图来设置这些权限条件的等效方法，则需使用下列语句：

```
grant all on titles to public
revoke select, update on titles (price, advance,
    total_sales)
from public
grant select, update on titles (price, advance,
    total_sales)
to sales
```

第二种方法可能遇到的一个问题是如果用户不在 sales 组中而输入了 select * from titles 命令，则此用户可能会意外地看到包含以下短语的消息：

```
permission denied
```

Adaptive Server 将星号扩展到 titles 表中所有列的列表中，且由于已撤消非销售部门用户对某些列的权限，因此会拒绝对这些列的访问。错误消息列出了用户无权访问的所有列。

若要查看用户有权访问的所有列，非销售部门的用户必须要对它们进行明确指定。因此，较好的解决方法是创建视图并授予相应权限。

视图也可用于**上下文相关的保护**。例如，可以创建一个视图，仅授予数据录入员访问自己添加或更新的那些行的权限。要完成上述操作，必须在表中添加一列。用户每输入一行在此列中就自动缺省记录此用户的 ID。可以在 create table 语句中定义此缺省值，如下所示：

```
create table testtable
    (empid       int,
     startdate   datetime,
     username     varchar(30) default user)
```

然后，定义一个包括表中所有行的视图。其中， uid 是当前用户：

```
create view context_view
as
    select *
    from testtable
    where username = user_name()
with check option
```

是否可以通过此视图检索行取决于在视图中发出 select 命令的用户的标识。将 with check option 添加到视图定义中之后，即可防止数据录入员在 username 列中伪造信息。

使用存储过程作为安全机制

如果同一个用户同时拥有存储过程与所有底层的对象，那么他可以授予其他用户使用存储过程的权限而不用授予对底层对象的权限。例如，可以授予用户执行存储过程 （更新指定表上行列子集）的权限，尽管用户没有对此表的任何其它权限。

角色和存储过程

使用 grant execute 命令将存储过程的 execute 权限授予具有特定角色的所有用户。 revoke execute 可删除此权限。但是， grant execute 权限不会妨碍对没有指定角色的用户授予存储过程的 execute 权限。

为进一步增加安全性，可以通过在存储过程中使用 has_role 系统函数对此过程的使用予以限制，保证某一过程只能由具有给定角色的用户执行。如果用户具有特定角色 （sa_role、 sso_role、 oper_role 或任何用户定义角色），则 has_role 返回 1，如果用户不具有该角色，则返回 0。例如，下面是一个使用 has_role 查看用户是否具有系统管理员角色的过程：

```
create proc test_proc
as
if (has_role("sa_role") = 0)
begin
    print "You don't have the right role"
    return -1
end
else
    print "You have SA role"
    return 0
```

有关 has_role 的详细信息，请参见 《参考手册：构件块》中的 “系统函数”。

了解所有权链

视图可依据其它视图或表而形成。过程可依据其它过程、视图或表而形成。这种依赖性可以看作是所有权链。

通常，视图所有者还拥有此视图的基础对象 （其它视图和表），存储过程所有者拥有此过程引用的所有过程、表和视图。

同存储过程及其引用的所有对象一样，视图及其基础对象常常在同一数据库中；但并不要求必须在同一数据库中。如果对象在不同的数据库中，则要使用视图或存储过程的用户必须是所有包含这些对象的数据库的有效用户或 guest 用户。这样，除非数据库所有者已授权，否则用户无法访问数据库。

某个用户，如果已具有某一过程或视图的 execute 权限，则在使用此视图或过程时， Adaptive Server 在下列情况下不检查任何基础对象的权限：

- 这些对象和视图或过程由同一用户拥有，并且
- 访问视图或过程的用户在每个包含基础对象的数据库中都是有效用户或 guest 用户。

但是，如果并非所有对象都由同一用户拥有，则 Adaptive Server 将在所有权链断开时检查对象权限。也就是说，如果对象 A 引用对象 B，而对象 B 不是由拥有对象 A 的用户拥有，则 Adaptive Server 会检查对象 B 的权限。使用这种方法， Adaptive Server 允许原始数据所有者对有权访问此数据的用户进行控制。

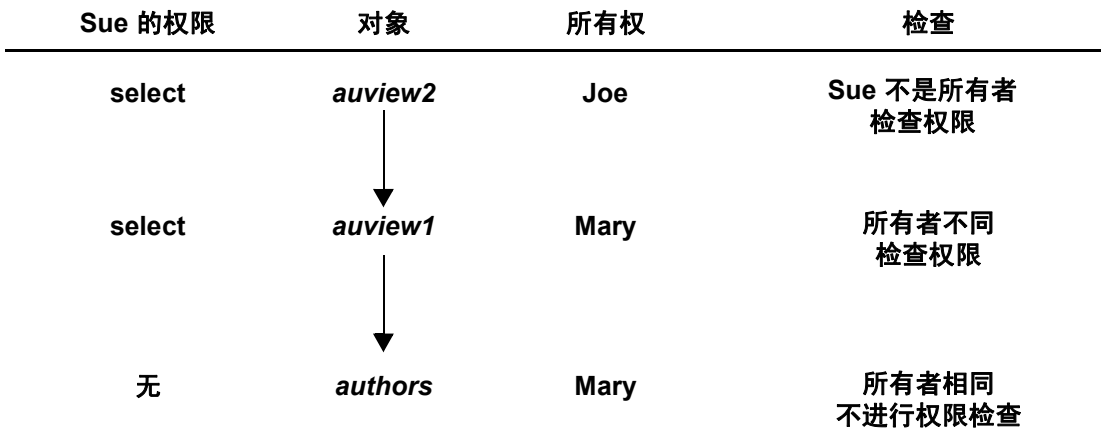
通常，创建视图的用户只需关注此视图的授权问题。例如，假设 Mary 已在其拥有的 authors 表中创建了名为 auview1 的视图。如果 Mary 将 auview1 的 select 权限授予 Sue，则 Adaptive Server 将允许 Sue 访问它，而不检查她对 authors 的权限。

但是，某个用户如果要依据另一用户所属对象创建一个视图或存储过程，则此用户必须知道自己的任何授权都取决于这些对象所有者允许的权限。

视图和所有权链的示例

假设 Joe 依据 Mary 的视图 auview1 创建了名为 auview2 的视图。然后，Joe 授予 Sue 对 auview2 的 select 权限。

图 6-2：视图的所有权链和权限检查，示例 1

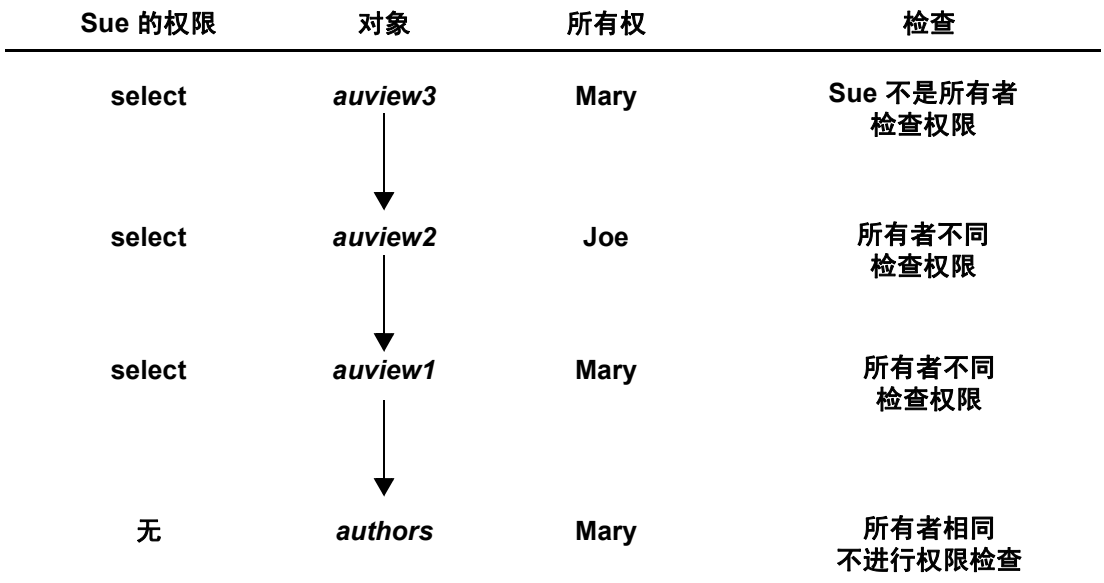


Adaptive Server 检查 auview2 和 auview1 的权限，发现 Sue 可以使用它们。 Adaptive Server 检查 auview1 和 authors 的所有权，发现它们的所有者相同。因此， Sue 可以使用 auview2。

将此例进行一下扩展，假设 Joe 的视图 auview2 依据 auview1 而创建，而视图 auview1 又依据 authors 而创建。Mary 认为自己喜欢 Joe 的 auview2，因此以其为基础创建了 auview3。这样， auview1 和 authors 的所有者都为 Mary。

所有权链如下图所示：

图 6-3：视图的所有权链和权限检查，示例 2

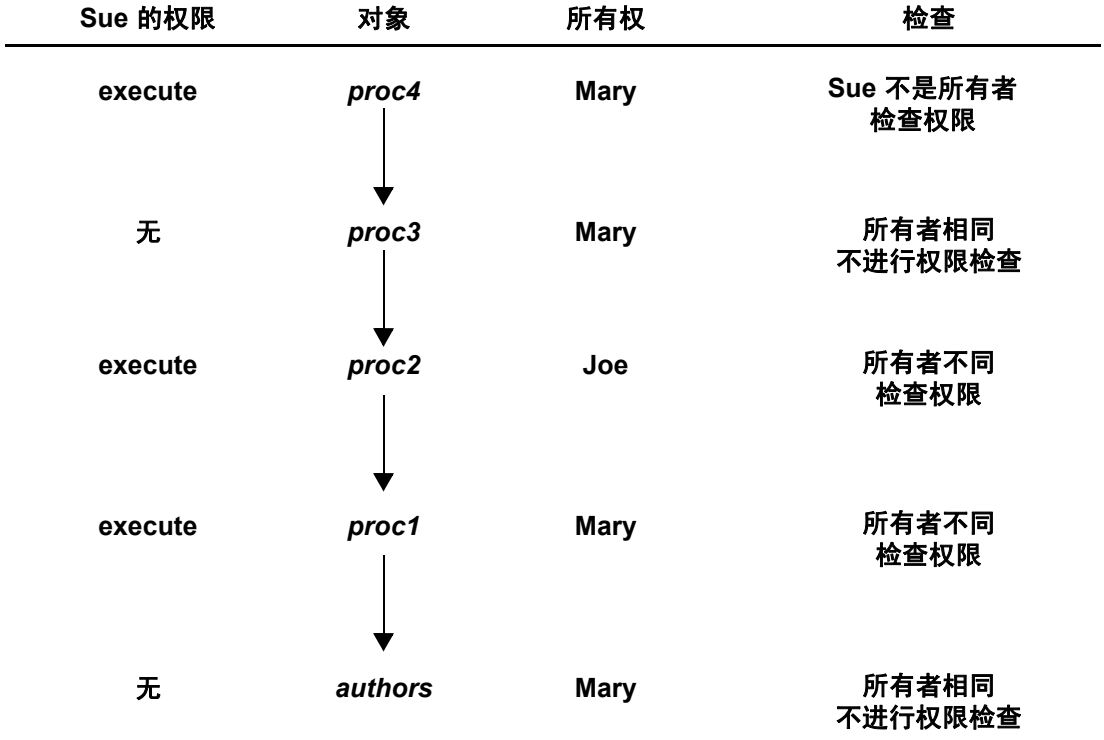


当 Sue 试图访问 auview3 时， Adaptive Server 检查 auview3、 auview2 和 auview1 的权限。如果 Joe 已授予 Sue auview2 的权限，Mary 也已授予她 auview3 和 auview1 的权限，则 Adaptive Server 将允许进行访问。Adaptive Server 仅在此链中某对象的前面一个对象具有不同所有者时 （或者，如果对象是此链中第一个对象时）才检查权限。例如，它检查 auview2，因为此视图的前一个对象 auview3 由不同的用户拥有。它不检查 authors 的权限，因为直接依赖于此视图的对象 auview1 由同一用户拥有。

过程和所有权链的示例

过程遵循的规则与视图相同。例如，假设所有权链如下所示：

图 6-4：存储过程的所有权链和权限检查



若要执行 proc4， Sue 必须要有执行 proc4、 proc2 和 proc1 的权限。而执行 proc3 的权限不是必需的，因为 proc3 和 proc4 的所有者相同。

Sue 每次执行 proc4 时， Adaptive Server 都要检查她对 proc4 及其引用的所有对象的权限。 Adaptive Server 知道要检查哪些引用对象：在 Sue 首次执行 proc4 时，它就已经确定了要检查的引用对象，并将此信息同过程执行计划一同保存。除非此过程引用的对象中有一个对象被删除或重新定义，否则， Adaptive Server 不会更改初始确定的检查对象内容。

使用此保护层次，每个对象所有者可以完全控制对对象的访问。所有者可以控制对视图、存储过程和表的访问。

触发器的权限

触发器是一种特殊的存储过程，用于强制实现完整性，尤其是参照完整性。触发器从来都不能直接执行，而仅作为修改表的副作用。不得 grant 或 revoke 触发器的权限。

只有对象所有者才可创建触发器。但是，如果表中的触发器引用不同用户拥有的对象，则可能中断所有权链。适用于过程的保护层次规则也适用于触发器。

如果受触发器影响的对象通常由拥有此触发器的用户所有，则可以编写一个触发器，用来修改由另一用户拥有的对象。这种情况下，任何以激活此触发器的方法修改对象的用户都必须具有另一对象的权限。

如果 Adaptive Server 由于触发器影响用户对其没有权限的对象而拒绝授予对数据修改命令的权限，则整个数据修改事务会回滚。

请参见 《Transact-SQL 用户指南》中的第 19 章 “触发器：强制实施参照完整性”。

使用行级访问控制

行级访问控制可提供以下功能，使数据库所有者或表所有者能够自动创建安全数据访问环境：

- 更加细化的数据安全性：可针对个别的行设置权限，而不仅仅针对表和列
- 根据组、角色和应用程序进行的自动数据过滤
- 服务器中编码的数据级安全性

行级访问控制通过三项功能限制对表的个别行中数据的访问：

- 数据库所有者定义并绑定到表的访问规则
- 应用程序环境功能，可提供定义、存储和检索用户定义环境的内置函数
- 数据库所有者、 sa_role 或用户可以创建的登录触发器

Adaptive Server 为所有数据操纵语言 (DML) 实施了行级访问控制，从而可以防止用户绕过访问控制来获取数据。

配置系统实施行级访问控制的语法如下：

```
sp_configure "enable row level access", 1
```

此选项会稍微增加 Adaptive Server 使用的内存量，并且需要 ASE_RLAC 许可证选项。行级访问控制是一个动态选项，因此不需要重新启动 Adaptive Server。

访问规则

若要使用行级访问控制功能，请将 access 选项添加到现有的 create rule 语法中。访问规则将对任何可被查看或修改的行施加限制。

访问规则类似于域规则，域规则允许表所有者控制用户可以在某个列中插入或更新的值。域规则对添加的数据施加限制，对 update 和 insert 命令起作用。

访问规则对检索的数据施加限制，对 select, update 和 delete 操作起作用。Adaptive Server 将对查询所读取的所有列强制执行访问规则，即使 select 列表中不包括这些列。也就是说，在给定的查询中， Adaptive Server 将对更新的表强制执行域规则，并对所读取的所有表强制执行访问规则。

例如：

```
insert into orders_table
select * from old_orders_table
```

在此查询中，如果 orders_table 上有域规则， old_orders_table 上有访问规则，则 Adaptive Server 将对 orders_table 强制执行域规则 （原因是更新了该表），而对 old_orders_table 强制执行访问规则 （原因是读取了该表）。

使用访问规则类似于使用视图或带有 where 子句的即席查询。查询是在附加访问规则之后才进行编译和优化的，因此它不会导致性能下降。访问规则提供表数据的虚拟视图，视图的内容取决于绑定到列的具体访问规则。

访问规则可以绑定到用 sp_addtype 定义的用户定义数据类型。 Adaptive Server 在用户表上强制执行访问规则，这样，表所有者或数据库所有者在标准化方案中就无需执行将访问规则绑定到列的维护任务。例如，您可以创建一个名为 username 的用户定义类型，其基类型为 varchar(30)，并为该类型绑定一个访问规则。 Adaptive Server 将对应用程序中任何具有 username 类型的列的表强制执行访问规则。

如第 204 页的 “访问规则作为用户定义的 Java 函数”和第 208 页的 “使用应用程序环境功能”所述，应用程序开发人员可以使用 Java 和应用程序环境编写灵活的访问规则。

访问规则的语法

在 create rule 语法中使用 access 参数创建访问规则。

```
create [or|and] access rule (access_rule_name)
as (condition)
```

创建带有访问规则的示例表

本节说明创建表并将访问规则绑定到该表的过程。

创建表

表所有者创建并填充表 T （username char(30)、 title char(30)、 classified_data char(1024)）：

```
AA, "Administrative Assistant","Memo to President"
AA, "Administrative Assistant","Tracking Stock
Movements"
VP1, "Vice President", "Meeting Schedule"
VP2, "Vice President", "Meeting Schedule"
```

创建并绑定访问规则

表所有者创建访问规则 uname_acc_rule 并将其绑定到表 T 上的 username 列。

```
create access rule uname_acc_rule
as @username = suser_name()
-----------
sp_bindrule uname_acc_rule, "T.username"
```

查询表

发出以下查询后：

```
select * from T
```

Adaptive Server 处理绑定到表 T 上 username 列的访问规则并将其附加到查询树。然后优化查询树，并生成和执行查询计划，就好像用户执行带有访问规则中给定的过滤器子句一样。也就是说， Adaptive Server 按如下示例附加访问规则并执行查询：

```
select * from T where T.username = suser_name().
```

服务器强制执行条件 `where T.username = suser_name()`。用户不能绕过访问规则。

管理助手执行选择查询的结果是：

```
AA, "Administrative Assistant","Memo to President"
AA, "Administrative Assistant","Tracking Stock
Movements"
```

删除访问规则

在删除访问规则之前，必须使用 sp_unbindrule 取消它与任何列或数据类型的绑定，如下面的示例所示：

```
sp_unbindrule "T.username",
NULL, "all"
```

sp_unbindrule 可取消绑定缺省情况下附加到列上的任何域规则。

在取消绑定规则之后，可以将其删除：

```
drop rule "rule_name"
```

例如：

```
drop rule "T.username"
```

扩展访问规则的语法

每个访问规则绑定到一个列，但一个表中可以具有多个访问规则。create rule 提供 AND 和 OR 参数来处理多个访问规则的计算。若要创建 AND 访问规则和 OR 访问规则，请使用扩展访问规则语法：

- AND 访问规则：

  create and access rule rule_name

- OR 访问规则：

  create or access rule rule_name as

可以将 AND 访问规则和 OR 访问规则绑定到列或用户定义数据类型。虽然每个列只能绑定一个访问规则，但通过扩展访问规则语法，可以在表中绑定多个访问规则。当访问表时，这些访问规则就会生效，缺省情况下，先执行绑定的 AND 规则，然后执行 OR 访问规则。

如果在表中绑定多个访问规则但没有定义 AND 或 OR 访问，则缺省访问规则是 AND。

如果表中某行上只有一个访问规则，并且这一规则被定义为 OR 访问规则，则它与 AND 访问规则的行为相同。

使用访问规则和扩展访问规则

创建访问规则

以下步骤创建访问规则：

```
create access rule empid1_access
as @empid = 1

create access rule deptno1_access
as @deptid = 2
```

以下步骤创建 OR 访问规则：

```
create or access rule name1_access
as @name = "smith"

create or access rule phone_access
as @phone = "9999"
```

创建表

此步骤创建一个测试表：

```
create table testtab1 (empno int, deptno int,name
char(10), phone char(4))
```

将规则绑定到表

以下步骤将访问规则绑定到测试表中的列：

```
sp_bindrule empid1_access, "testtab1.empno"
/*Rule bound to table column.*/
(return status = 0)

sp_bindrule deptno1_access,"testtab1.deptno"
/*Rule bound to table column.*/

(return status = 0)

sp_bindrule name1_access,"testtab1.name"
/*Rule bound to table column.*/

(return status = 0)

sp_bindrule phone_access,"testtab1.phone"
/*Rule bound to table column.*/

(return status = 0)
```

将数据插入表中

以下步骤将值插入测试表中：

```
insert testtab1 values (1,1,"smith","3245")
(1 row affected)

insert testtab1 values(2,1,"jones","0283")
(1 row affected)

insert testtab1 values(1,2,"smith","8282")(1 row
affected)

insert testtab1 values(2,2,"smith","9999")

(1 row affected)
```

访问规则示例

以下示例说明访问规则如何返回包含访问规则限制的信息的特定行。

示例 1. 返回两行中的信息：

```
/* return rows when empno = 1 and deptno = 2
and ( name = "smith" or phone = "9999" )
*/

select * from testtab1
 empno       deptno      name        phone
------------ ----------- ---------- -----
             1           2 smith      8282
             1           2 jones      9999

(2 rows affected)

/* unbind access rule from specific column */
sp_unbindrule "testtab1.empno",NULL,"accessrule"
/*Rule unbound from table column.*/

 (return status = 0)
```

示例 2. 返回四行中的信息：

```
/* return rows when deptno = 2 and ( name = "smith"
or phone = "9999" )*/

select * from testtab1

 empno       deptno      name        phone
----------- ----------- ---------- -----
1            2           smith      8282
2            2           smith      9999
3            2           smith      8888
1            2           jones      9999

(4 rows affected)

/* unbind all deptno rules from specific column */
sp_unbindrule "testtab1.deptno",NULL,"all"
/*Rule unbound from table column.*/

(return status = 0)
```

示例 3. 返回六行中的信息：

```
/* return the rows when name = "smith" or phone = "9999"
*/

select * from testtab1
 empno       deptno      name       phone
 ----------- ----------- ---------- -----
           1           1 smith      3245
           1           2 smith      8282
           2           2 smith      9999
           3           2 smith      8888
           1           2 jones      9999
           2           3 jones      9999
```

访问规则和 alter table 命令

如果表所有者使用 alter table 命令，则在执行此命令期间， Adaptive Server 将禁用访问规则并在命令完成后启用它们。在执行 alter table 命令期间，禁用访问规则是为了避免过滤表数据。

访问规则和 *bcp*

使用 bcp 从表中向外复制数据时， Adaptive Server 将强制执行访问规则。由于对表具有选择权限的任何用户都可使用 bcp，因此 Adaptive Server 不能像对 alter table 那样禁用访问规则。

为了安全目的，批量向外复制期间，数据库所有者应以独占方式锁定表并禁用访问规则。禁用访问规则时，此锁定禁止其他用户访问。复制完数据之后，数据库所有者应绑定访问规则并将表解锁。

访问规则作为用户定义的 Java 函数

访问规则可以使用用户定义的 Java 函数。例如，可以使用 Java 函数编写复杂的规则，在规则中使用应用程序的配置文件、登录到应用程序的用户和当前分配给用户的应用程序的角色。

以下 Java 类使用 GetSecVal 方法来演示如何在访问规则中使用将 JDBC 用作用户定义函数的 Java 方法：

```
import java.sql.*;
import java.util.*;

public class sec_class {
```

```
static String _url = "jdbc:sybase:asejdbc";
public static int GetSecVal(int c1)
{
try
{
PreparedStatement pstmt;
ResultSet rs = null;
Connection con = null;
    int pno_val;

pstmt = null;

Class.forName("sybase.asejdbc.ASEDriver");
con = DriverManager.getConnection(_url);

if (con == null)
{
return (-1);
}

pstmt = con.prepareStatement("select classification
from sec_tab where id = ?");

if (pstmt == null)
{
return (-1);
}

pstmt.setInt(1, c1);

rs = pstmt.executeQuery();

rs.next();

pno_val = rs.getInt(1);

rs.close();

pstmt.close();

con.close();

return (pno_val);

}
catch (SQLException sqe)
```

```
{
return(sqe.getErrorCode());
}
catch (ClassNotFoundException e)
{

System.out.println("Unexpected exception :" +
e.toString());
System.out.println("\nThis error usually indicates that
" + "your Java CLASSPATH environment has not been set
properly.");
e.printStackTrace();
return (-1);
}
catch (Exception e)
{
System.out.println("Unexpected exception :" +
e.toString());
e.printStackTrace();
return (-1);
}
}
}
```

在编译完 Java 代码后，可以从 isql 运行相同的程序，如下所示。

例如：

```
javac sec_class.java
jar cufo sec_class. jar sec_class.class
installjava -Usa -Password -
f/work/work/FGAC/sec_class.jar -
-D testdb
```

在 isql 中：

```
/*to create new user datatype class_level*/
sp_addtype class_level, int
/*to create the sample secure data table*/
create table sec_data (c1 varchar(30),
c2 varchar(30),
c3 varchar(30),
clevel class_level)
/*to create the classification table for each user*/
create table sec_tab (userid int, clevel class-level
int)

insert into sec_tab values (1.10)
```

```
insert into sec_tab values (2.9)
insert into sec_tab values (3.7)
insert into sec_tab values (4.7)
insert into sec_tab values (5.4)
insert into sec_tab values (6.4)
insert into sec_tab values (7,4)

declare @v1 int
select @v1 = 5
while @v1 > 0
begin
insert into sec_data values('8', 'aaaaaaaaaa',
'aaaaaaaaaa', 8)
insert into sec_data values('7', 'aaaaaaaaaa',
'aaaaaaaaaa', 7)
insert into sec_data values('5', 'aaaaaaaaaa',
'aaaaaaaaaa', 5)
insert into sec_data values('5', 'aaaaaaaaaa',
'aaaaaaaaaa', 5)
insert into sec_data values('2', 'aaaaaaaaaa',
'aaaaaaaaaa', 2)
insert into sec_data values('3', 'aaaaaaaaaa',
'aaaaaaaaaa', 3)
select @v1 = @v1 -1
end
go

create access rule clevel_rule
@clevel <= sec_class.GetSecVal(suser_id())
go

create default clevel_def as
sec_class.GetSecVal(suser_id())
go

sp_bindefault clevel_def, class_level
go

sp_bindrule clevel, class_level
go

grant all on sec_data to public
go
grant all on sec_tab to public
go
```

使用应用程序环境功能

数据库服务器上的应用程序必须限制对数据的访问。在编写应用程序代码时应仔细考虑用户的配置文件。例如，编写人力资源应用程序时需要了解应允许哪些用户更新工资信息。

实现这种编码方式的各种属性构成了应用程序环境。应用程序环境功能 (ACF) 由三个内置函数组成，通过允许访问规则与会话中分配给用户的固有值进行比较，它们可以提供安全的数据访问环境。

应用程序环境包括 context_name、 attribute_name 和 attribute_value。用户为每个环境定义环境名、属性和值。可以使用 Sybase 提供的缺省只读应用程序环境 SYS_SESSION 来访问某些会话特定的信息。此应用程序环境在第 214 页的表 6-4 中进行了介绍。也可以创建自己的应用程序环境，如第 210 页的 “创建和使用应用程序环境”所述。

用户配置文件和应用程序配置文件一起提供了附加的和叠加的安全方案，应用程序配置文件在由系统管理员创建的表中定义。

ACF 允许用户定义、存储和检索：

- 用户配置文件 （授权给用户的角色和用户所属的组）
- 当前使用的应用程序配置文件

每个会话可能会有任意数目的应用程序环境，任何环境都可以定义任意数目的属性/值对。 ACF 环境行因不同的会话而异，不能在多个会话间保持；但是，与本地变量不同的是，它们可以用于不同的语句执行嵌套级别。 ACF 提供了用于设置、获取、列出和删除这些环境行的内置函数。

设置使用应用程序环境函数的权限

可以在选择语句中执行应用程序环境函数。函数所有者是服务器的系统管理员。可以使用内置函数创建、设置、检索和删除应用程序环境。

函数中使用的数据是在包含所有表的所有登录的表中定义的，该表由系统管理员创建。有关此表的详细信息，请参见第 215 页的 “使用登录触发器”。

- set_appcontext() 存储：

```
select set_appcontext ("titles", "rlac", "1")
```

- get_appcontext() 在会话中提供环境的两部分，并检索第三部分：

```
select get_appcontext ("titles", "rlac")
------------------------
1
```

有关这些函数、 list_appcontext 和 rm_appcontext 的详细信息，请参见第 210 页的 “创建和使用应用程序环境”。

授予和撤消

授予和撤消给定数据库中的用户、角色和组访问该数据库中对象的特权。但也有例外，如 create database、 set session authorization 和 connect。授予了这些特权的用户应该是 master 数据库中的有效用户。若要使用其它特权，用户必须是对象所在的数据库中的有效用户。

函数的使用意味着，除非有特殊安排，否则任何登录用户都可以重置会话的配置文件。虽然 Adaptive Server 会审计内置函数，但在发现问题前还是会受到安全威胁。若要限制对这些内置函数的访问，请使用 grant 和 revoke 权限。只有具有 sa_role 的用户才能授予或撤消对这些内置函数的权限。作为这些函数执行的在服务器上强制实施的数据访问控制检查的一部分，将只检查 select 权限。

有效用户

函数没有对象 ID，也没有主数据库。因此，每个数据库所有者必须向适当的用户授予这些函数的选择权限。 Adaptive Server 会查找用户的缺省数据库并检查对此数据库的权限。使用此方法，只有用户的缺省数据库的所有者才需要授予选择特权。如果要限制其它数据库，则相应数据库的所有者必须显式地撤消用户在那些数据库中的权限。

当授予和撤消对应用程序环境内置函数的特权时，只有这些函数会对用户执行数据访问控制检查。授予或撤消其它函数的特权在 Adaptive Server 中没有任何影响。

授予 public 的特权只影响在系统管理员创建的表中命名的用户。有关该表的信息，请参见第 215 页的 “使用登录触发器”。只有在 sa_role 将 Guest 用户添加到表中明确授予某些权限后，Guest 用户才具有相应权限。

系统管理员可以执行以下命令，来授予或撤消对特定应用程序环境函数的 select 特权。

- grant select on set_appcontext to user_role
- grant select on set_appcontext to joe_user
- revoke select on set_appcontext from joe_user

创建和使用应用程序环境

以下函数可用于创建和维护应用程序环境。有关详细信息，请参见《参考手册：构件块》。

- set_appcontext
- get_appcontext
- list_appcontext
- rm_appcontext

set_appcontext

为指定的用户会话设置由应用程序的属性定义的应用程序环境名、属性名和属性值。

set_appcontext ("*context_name*", "*attribute_name*", "*attribute_value*")

参数

- *context_name* — 指定应用程序环境名的行，作为 char(30) 数据类型保存。
- *attribute_name* — 指定应用程序环境属性名的行，作为 char(30) 数据类型保存。
- *attribute_value* — 指定应用程序属性值的行，作为 char(255) 数据类型保存。

示例

示例 1. 创建名为 CONTEXT1 的应用程序环境，其属性 ATTR1 的值为 VALUE1：

```
select set_appcontext("CONTEXT1", "ATTR1", "VALUE1")
---------------
0
```

示例 2. 演示尝试替换现有的应用程序环境。尝试失败，返回 -1：

```
select set_appcontext("CONTEXT1", "ATTR1", "VALUE1")
--------------
-1
```

示例 3. 演示 set_appcontext 如何在值中包括数据类型转换：

```
declare@val numeric
select @val = 20
select set_appcontext ("CONTEXT1", "ATTR2",
convert(char(20), @val))
------------
0
```

示例 4. 显示没有相应权限的用户尝试设置应用程序环境时所返回的结果。尝试失败，返回 -1：

```
select set_appcontext("CONTEXT1", "ATTR2", "VALUE1")
--------------
-1
```

用法

- set_appcontext 返回 0 表示成功，返回 -1 表示失败。
- 如果设置的值在当前会话中已存在，则 set_appcontext 返回 -1。
- set_appcontext 不能替换现有应用程序环境的值。若要为一个环境指定新值，应先删除该环境，然后使用新值重新创建一个环境。
- set_appcontext 将属性保存为 char 数据类型。如果您创建的访问规则必须将属性值与另一个数据类型进行比较，则该规则应该将 char 数据转换为适当的数据类型。
- 此函数中的所有参数都是必需的。

get_appcontext

返回指定环境中的属性值。

get_appcontext ("*context_name*", "*attribute_name*")

参数

- *context_name* — 指定应用程序环境名的行，作为 char(30) 数据类型保存。
- *attribute_name* — 指定应用程序环境属性名的行，作为 char(30) 数据类型保存。

示例

示例 1. 显示为 ATTR1 返回的 VALUE1：

```
select get_appcontext("CONTEXT1", "ATTR1")
-----------
VALUE1
```

示例 2. CONTEXT2 中不存在 ATTR1：

```
select get_appcontext("CONTEXT2", "ATTR1")
------------
NULL
```

示例 3. 显示没有相应权限的用户尝试获取应用程序环境时所返回的结果：

```
select get_appcontext("CONTEXT1", "ATTR2")
select permisssion denied on built-in get_appcontext,
database dbid
----------
-1
```

用法

- get_appcontext 返回 0 表示成功，返回 -1 表示失败。
- 如果应用程序环境中不存在所需的属性，则 get_appcontext 返回“null”。
- get_appcontext 将属性保存为 char 数据类型。如果您创建的访问规则需将属性值与其它数据类型进行比较，则该规则应该将 char 数据转换为适当的数据类型。
- 此函数中的所有参数都是必需的。

list_appcontext

列出当前会话中所有环境的全部属性。

list_appcontext ("*context_name*")

参数

- *context_name* — 命名会话中的所有应用程序环境属性。list_appcontext 的数据类型为 char(30)。

示例

示例 1. 显示具有适当权限的用户列出应用程序环境的结果：

```
select list_appcontext ("*", "*")
Context Name:(CONTEXT1)
Attribute Name:(ATTR1) Value:(VALUE2)
Context Name:(CONTEXT2)
Attribute Name:(ATTR1) Value:(VALUE!)
-----------
0
```

示例 2. 显示没有适当权限的用户试图列出应用程序环境的结果。尝试失败，返回 -1。

```
select list_appcontext()
Select permission denied on built-in
list_appcontext, database DBID
---------
-1
```

用法

- list_appcontext 返回 0 表示成功，返回 -1 表示失败。
- 因为内置函数不会返回多个结果集，客户端应用程序收到 list_appcontext 以消息形式返回的结果。

权限

若要使用 list_appcontext，用户必须具有适当的权限。有关详细信息，请参见第 208 页的 “设置使用应用程序环境函数的权限”。

rm_appcontext

删除特定应用程序环境或所有应用程序环境。

rm_appcontext ("*context_name*", "*attribute_name*")

参数

- *context_name* — 指定应用程序环境名的行，作为 char(30) 数据类型保存。
- *attribute_name* — 指定应用程序环境属性名的行，作为 char(30) 数据类型保存。

示例

示例 1. 使用星号 ("*") 删除指定环境中的所有属性。

```
select rm_appcontext("CONTEXT1", "*")
---------
0
```

示例 2. 使用星号 ("*") 删除所有环境和属性。

```
select rm_appcontext("*", "*")
---------
0
```

示例 3. 显示用户试图删除不存在的环境。尝试失败，返回 -1。

```
select rm_appcontext("NON_EXISTING_CTX", "ATTR2")
---------
-1
```

示例 4. 显示没有相应权限的用户尝试删除应用程序环境时所返回的结果。

```
select rm_appcontext("CONTEXT1", "ATTR2")
---------
-1
```

用法

- rm_appcontext 返回 0 表示成功，返回 -1 表示失败。
- 此函数中的所有参数都是必需的。

SYS_SESSION 系统应用程序环境

SYS_SESSION 环境显示缺省的预定义应用程序环境，该环境提供特定于会话的属性/值对。使用该环境的语法是：

select list_appcontext ("SYS_SESSION", "*")

然后：

```
select get_appcontext ("SYS_SESSION", "<attribute>")
```

表 6-4：SYS_SESSION 属性和值

属性	值
username	登录名
hostname	客户端与其连接的主机的名称
applname	客户端设置的应用程序的名称
suserid	当前数据库中用户的用户 ID
groupid	当前数据库中用户的组 ID
dbid	用户的当前数据库的 ID
dbname	当前数据库
spid	服务器进程 ID
proxy_suserid	代理的服务器用户 ID
client_name	中间层应用程序使用 set client_name 设置的客户端名称
client_applname	中间层应用程序使用 set client_applname 设置的客户端应用程序名称
client_hostname	中间层应用程序使用 set client_hostname 设置的客户端主机名称
language	缺省情况下或使用 set language 设置后，客户端正在使用的当前语言 (@@language)
character_set	客户端正在使用的字符集 (@@client_csname)
dateformat	客户端需要的日期格式，使用 set dateformat 设置
is_showplan_on	如果 set showplan 处于启用状态，则返回 YES，如果处于禁用状态则返回 NO
is_noexec_on	如果 no exec 处于启用状态，则返回 YES，如果处于禁用状态则返回 NO

使用访问规则和 ACF 解决问题

本节将为如下问题提供一个解决方案：有 5 个用户处于不同的安全级别，每个用户应该只能看到相关值小于或等于该用户的安全级别的行。此解决方案使用访问规则，结合应用程序环境功能，只显示其中的一个用户 Dave 可看到的行。

有以下 5 个登录名：

- Anne 的安全级别为 1。
- Bob 的安全级别为 1。
- Cassie 的安全级别为 2。
- Dave 的安全级别为 2。
- Ellie 的安全级别为 4。

用户应该只能看到 rlac 中的值小于或等于各自的安全级别的行。若要实现此目的，可创建一个访问规则并应用 ACF。

rlac 列的类型为 integer， appcontext 参数的类型为 char。

```
create access rule rlac_rule as
  @value <= convert(int, get_appcontext("titles",
        "rlac"))

sp_bindrule rlac_rule, "titles.rlac"

/* log in as Dave and apply ACF value of 2*/

select set_appcontext("titles", "rlac", "2")

/*this value persists throughout the session*/
/*select all rows*/

select title_id, rlac from titles
----------------------
title_id     rlac
------------ --------
PC8888        1
BU1032        2
PS7777        1
PS3333        1
BU1111        2
PC1035        1
BU2075        2
PS2091        1
PS2106        1
BU7832        2
PS1372        1

(11 rows affected)
```

使用登录触发器

注释　本节中的一些信息摘自 http://www.sypron.nl/logtrig.html 中的文章 "Login Triggers in ASE 12.5" （ASE 12.5 中的登录触发器）。版权所有 1998– 2002， Rob Verschoor/ Sypron B.V.

登录触发器在每次用户登录时执行指定的存储过程。登录触发器是普通的存储过程，只不过它在后台执行。它是成功登录过程的最后一个步骤，并为登录的用户设置应用程序环境。

只有系统安全员才能向服务器中的用户注册登录触发器。

若要提供安全的环境，系统管理员必须：

1 撤消对 set_appcontext 函数的 select 权限。登录触发器的所有者必须明确地具有使用 set_appcontext 的权限，即使该所有者具有 sa_role 也需要如此。

2 利用存储过程为每个用户配置登录触发器，然后向用户注册登录触发器。

3 向用户执行的登录触发器提供执行特权。

创建登录触发器

以存储过程的形式创建登录触发器。不要使用 create trigger 命令。以下示例需要您先在 pubs2 数据库中创建 lookup 表：

```
create table lookup (
  appname varchar(20),
  attr varchar(20),
  value varchar(20),
  login varchar(20)
)
```

然后，在 pubs2 数据库中创建一个登录触发器存储过程：

```
create procedure loginproc as
  declare @appnamevarchar(20)
  declare @attr varchar(20)
  declare @value varchar(20)
  declare @retvalue int
declare apctx cursor for
select appname, attr, value from
pubs2.dbo.lookup where login = suser_name()
open apctx
fetch apctx into @appname, @attr, @value

While (@@sqlstatus = 0)
  begin
    select f@retval =
      set_appcontext (rtrim (@appname),
      rtrim(@attr), rtrim(@value))
  fetch apctx into @appname, @attr, @value
end
go
```

向 public 授予执行 loginproc 的权限：

```
grant execute on loginproc to public
```

若要将特定用户与登录触发器关联，请在该用户的缺省数据库中运行 alter login。

配置登录触发器

您必须启用 sso_role 才能设置、更改或删除登录触发器。登录触发器的对象 ID 存储在 syslogins.procid 列中。缺省情况下，登录触发器不存在。必须使用 alter login 对它们进行注册。

在用户的缺省数据库中运行此命令。作为登录触发器注册的存储过程在用户的缺省数据库中必须可用，原因是 Adaptive Server 将在用户的缺省数据库中搜索 sysobjects 表，以查找登录触发器对象。

配置登录触发器

下面的示例将存储过程 my_proc （它必须存在于您要配置的数据库中）配置为 Adaptive Server 登录名 my_login 的登录触发器：

```
alter login my_login modify login script "my_proc"
```

同样，必须在用户的缺省数据库中执行该命令。 Adaptive Server 将检查登录名是否对存储过程具有 execute 权限，但只有在用户实际登录并执行登录触发器时才会发生。

删除和更改登录触发器

一旦将存储过程配置为登录触发器，就不能删除该存储过程了。必须先取消对它的配置，要么连登录触发器一起删除，要么将登录触发器更改为其它存储过程。若要删除登录触发器，请输入：

```
alter login my_login drop login script
```

若要将登录触发器更改为其它存储过程，请输入：

```
alter login my_login modify login script "diff_proc"
```

显示登录触发器

若要显示当前登录触发器，请使用 sp_displaylogin：

```
sp_displaylogin my_login
go
(....)
Default Database:my_db
Default Language:
Auto Login Script:my_proc
....
```

执行登录触发器

登录触发器不同于普通的存储过程，原因在于它们一旦注册之后，便会在后台执行，而不需要活动的用户连接。在配置了登录触发器之后，只要用户一登录， Adaptive Server 就会在后台自动执行该登录触发器，并且执行登录触发器的过程会发生在服务器执行来自客户端应用程序的任何命令之前。

如果一个登录进行了多个并发连接，登录触发器会在每个会话中独立执行。同样，多个登录可以将同一个存储过程配置成登录触发器。

后台执行意味着您不能在配置为登录触发器的存储过程中使用存储过程的某些标准功能。例如，不能向该过程传送或从该过程接收任何没有缺省值的参数，该过程也不会传递回任何结果值。

这种特殊的执行模式会影响由登录触发器存储过程调用的任何存储过程，以及登录触发器存储过程本身生成的任何输出。

您还可以将登录触发器存储过程作为正常的存储过程执行，例如在 isql 命令中。该过程将正常地执行和发挥作用，像通常那样显示所有输出和错误消息。

了解登录触发器输出

将存储过程作为后台任务执行的主要影响是登录触发器的输出不会写入到客户端应用程序中，而是以一些 （但不是全部）错误消息的形式写入 Adaptive Server 错误日志文件。

在错误日志中，来自 print 或 raiserror 消息的输出的前缀为 background task message 或 background task error。例如，在 Adaptive Server 错误日志中，登录触发器中的语句 print "Hello!" 和 raiserror 123456 显示为：

```
(....) background task message:Hello!
(....) background task error 123456:This is test
message 123456
```

但是，并非所有输出都会写入 Adaptive Server 错误日志：

- select 语句 （通常被发送到客户端连接）的结果集不会出现在任何位置，甚至也不会出现在 Adaptive Server 错误日志中。此信息无处可寻。
- 通常执行以下语句：insert...select 和 select...into 语句，以及其它一些通常不将结果集发送到客户端应用程序的 DML 语句和存储过程中通常允许使用的 DDL 语句。

对其它应用程序使用登录触发器

登录触发器是作为 Adaptive Server 中行级访问控制功能的一部分设计的。在此环境中，一旦某个会话登录到 Adaptive Server 后，您就可以结合使用登录触发器与访问规则和应用程序环境的功能，设置行级访问控制。不过，还可以将登录触发器用于其它目的。

限制并发连接的数目

以下示例限制特定登录可与 Adaptive Server 建立的并发连接的数目。在需要被限制访问的用户的缺省数据库中执行示例中步骤 1 和 2 中描述的每个命令：

1　由系统管理员创建 limit_user_sessions 存储过程：

```
create procedure limit_user_sessions
as
  declare @cnt int,
   @limit int,
   @loginname varchar(32)

  select @limit = 2 – max nr. of concurrent logins

/* determine current #sessions */
select @cnt = count(*)
from master.dbo.sysprocesses
  where suid = suser_id()

/* check the limit */
if @cnt > @limit
begin
  select @loginname = suser_name()
  print "Aborting login [%1!]:exceeds session
    limit [%2!]",
      @loginname, @limit
    /* abort this session */
    select syb_quit()
end
go

grant exec on limit_user_sessions to public
 go
```

2　以系统安全员身份将此存储过程配置为用户 “bob”的登录触发器：

```
alter login bob modify login script
"limit_user_sessions"
go
```

3 现在，当用户 “bob”创建 Adaptive Server 的第三个会话时，登录触发器将调用 syb_quit() 函数终止该会话：

```
% isql -SASE125 -Ubob -Pbobpassword
1> select 1
2> go

CT-LIBRARY error:
ct_results():network packet layer:internal net
library error:Net-Library operation terminated due
to disconnect
```

4 此消息将显示在 Adaptive Server 错误日志文件中：

```
(...) background task message:Aborting login
[ my_login]:exceeds session limit [2]
```

强制执行基于时间的限制

此示例描述系统管理员如何通过创建登录触发器来对用户会话强制执行基于时间的限制。在需要被限制访问的用户的缺省数据库中执行步骤 1 – 4 中描述的每个命令：

1 以系统管理员身份创建此表：

```
create table access_times (
suid int not null,
dayofweek tinyint,
shiftstart time,
shiftend time)
```

2 由系统管理员将以下行插入表 access_times。这些行指示允许用户 “bob”在星期一上午 9 点到下午 5 点登录到 Adaptive Server，并允许用户 “mark”在星期二上午 9 点到下午 5 点登录到 Adaptive Server

```
insert into access_times
select suser_id('bob'), 1, '9:00', '17:00'
go
insert into access_times
select suser_id('mark'), 2, '9:00', '17:00'
go
```

3 以系统管理员身份创建 limit_access_time 存储过程，该存储过程引用 access_time 表来确定是否应授予登录访问权限：

```
create procedure limit_access_time as
declare @curdate date,
    @curdow tinyint,
    @curtime time,
    @cnt int,
    @loginname varchar(32)
```

```
-- setup variables for current day-of-week, time
select @curdate = current_date()
select @curdow = datepart(cdw,@curdate)
select @curtime = current_time()
select @cnt = 0

-- determine if current user is allowed access
select @cnt = count(*)
from access_times
where suid = suser_id()
and dayofweek = @curdow
and @curtime between shiftstart and shiftend

if @cnt = 0
begin
   select @loginname = suser_name()
   print "Aborting login [%1!]:login attempt past
     normal working hours", @loginname

   -- abort this session
   return -4
end
go

grant exec on limit_access_time to public
go
```

4　以系统安全员身份将 limit_access_time s存储过程配置为用户 “bob” 和 “mark” 的登录触发器：

```
alter login bob login script
"limit_access_time"
 go
alter login mark login script
"limit_access_time"
 go
```

5　在星期一，用户 “bob” 可以成功创建一个会话：

```
isql -Ubob -Ppassword
1> select 1
2> go
-----------
          1
(1 row affected)
```

但是，用户 “mark”无权访问 Adpative Server：

```
isql -Umark -Ppassword
1> select 1
2> go
CT-LIBRARY error:
ct_results():network packet layer:internal net
library error:Net-Library operation terminated
due to disconnect
```

6 将在错误日志中记录以下消息：

```
(...) server back-ground task message:Aborting
login [mark]:login attempt past normal working
hours
```

上述示例显示了如何限制特定登录名的并发连接的数目，以及如何限制该登录名在特定的时间进行访问，但它有一个缺点：客户端应用程序无法方便地检测到会话被终止的原因。若要向用户显示一条消息 （例如 “现在用户太多 — 请稍后重试”），请使用其它方法。

您可以在存储过程中故意引发一个错误以中止登录触发器存储过程，而不是调用内置函数 syb_quit()，该函数将使服务器仅仅是终止当前会话。

例如，用零作除数可以中止登录触发器存储过程、终止会话并导致出现消息。

登录触发器限制

以下操作受到限制。

- 不能创建 #temp 表在会话中稍后使用。过程一旦完成，就将自动删除 #temp 表并恢复初始会话设置，就像在任何其它存储过程中一样。
- 不要对 sa 登录名使用登录触发器；失败的登录触发器会将您锁定在 Adaptive Server 之外。
- 不要对任何执行时间超过几秒的操作或执行过程中可能存在问题的操作使用登录触发器。

发布和信息

- 如果不能访问 Adaptive Server 错误日志，请不要使用登录触发器。始终检查 Adaptive Server 错误日志中是否有错误消息。
- 对于 Adaptive Server 15.0.2 版及更高版本，当服务器启动时，在登录触发器中设置或取消设置的任何可导出选项将在登录过程中生效。

  若要禁用此行为，请在登录触发器内执行 set export_options off。

  Adaptive Server 15.0.1、 12.5.4 版及早期版本要求您使用跟踪标志 4073 启动 Adaptive Server 来启用登录触发器选项。
- 客户端应用程序 （如 isql）不知道登录触发器的存在或执行；它会在成功登录后立即显示命令提示符，虽然 Adaptive Server 在登录触发器成功执行前不会执行任何命令。即使登录触发器终止了用户连接，也会显示该 isql 提示符。
- 登录到 Adaptive Server 的用户必须具有 execute 权限才能使用登录触发器存储过程。如果未授予 execute 权限，则 Adaptive Server 错误日志中会显示一条错误消息并且用户连接将立即关闭 （虽然 isql 仍显示命令提示符）。

  Adaptive Server 错误日志显示类似于以下内容的消息：

  ```
  EXECUTE permission denied on object my_proc,
  database my_db, owner dbo
  ```
- 登录触发器存储过程不可以包含没有指定缺省值的参数。如果存储过程中出现不带缺省值的参数，则登录触发器将失败，并且 Adaptive Server 错误日志中出现类似于以下内容的错误：

  ```
  Procedure my_proc expects parameter @param1, which
  was not supplied...
  ```

禁用对登录触发器的执行特权

数据库所有者或管理员可以禁用对登录触发器的 execute 特权，或者对登录触发器进行编码，以便只允许特定时间内的访问。例如，您可能要禁止常规用户在数据库所有者或管理员更新表时使用数据库。

注释　如果登录触发器返回负数，则登录失败。

从登录触发器导出 set 选项

Adaptive Server 允许登录触发器内的 set 命令选项在整个用户会话期间保持有效。

将自动导出以下 set 选项：

- showplan
- arithabort [overflow | numeric_truncation]
- arithignore [overflow]
- colnames
- format
- statistics io
- procid
- rowcount
- altnames
- nocount
- quoted_identifier
- forceplan
- fmtonly
- close on endtran
- fipsflagger
- self_recursion
- ansinull
- dup_in_subquery
- or_strategy
- flushmessage
- ansi_permissions
- string_rtruncation
- prefetch
- triggers
- replication

- sort_resources
- transactional_rpc
- cis_rpc_handling
- strict_dtm_enforcement
- raw_object_serialization
- textptr_parameters
- remote_indexes
- explicit_transaction_required
- statement_cache
- command_status_reporting
- proc_return_status
- proc_output_params

设置全局登录触发器

可以使用 sp_logintrigger 设置在每个用户登录时执行的全局登录触发器。若要采取用户特定操作，请使用 alter login 或 create login 设置用户特定登录触发器。

注释　可通过设置跟踪标志 -T4073 来激活此选项。

第 7 章

数据的保密性

本章介绍如何配置 Adaptive Server 以确保所有数据都是安全和保密的。

Adaptive Server 中的安全套接字层 (SSL)

Adaptive Server Enterprise 安全服务现在支持安全套接字层 (SSL) 的基于会话的安全性。 **SSL** 是一种用于保护通过 Internet 传输敏感信息 （例如信用卡号、股票交易和银行交易）的标准。

虽然全面讨论公开密钥密码术超出该文档的范围，但是基本的信息值得说明一下，以便您对 SSL 如何保护 Internet 通信信道有所了解。该文档不是一个关于公开密钥密码术的综合指南。

Adaptive Server SSL 功能的实现基于以下假定：有一名知识渊博的系统安全员，他熟悉安全性策略和您节点的需要，以及对 SSL 和公用密钥密码术有一般性的了解。

因特网通信概述

TCP/IP 是在客户端/服务器计算中用到的主要传输协议，也是管理因特网上数据传输的协议。TCP/IP 使用中间的计算机将数据从发送者传输到接收者。中间计算机将易于篡改、偷窃、偷听和模拟的不安全连接引入通信系统。

公开密钥密码术

现在人们已经开发出几种机制并用以保护通过 Internet 传输的敏感数据，这些机制统称为**公开密钥密码术**。公开密钥密码术由加密、键交换、数字签名和数字认证组成。

加密

加密是使用密码算法对信息进行编码的过程，以防指定接收者以外的人利用这些信息。用于加密的密钥类型有两种：

- **对称密钥加密** — 用相同算法 （密钥）来对消息进行加密和解密。这种形式的加密安全性最差，由于密钥非常简单，因此也就容易解译。不过，因为这种形式加密和解密消息所需的计算最少，所以使用对称密钥加密的数据的传输速度很快。
- **公开/私有密钥加密** — 也称作非对称密钥，是由公开和私有组件组成、用来对消息进行加密和解密的一对密钥。虽说可能有所不同，但通常来说，消息都是由发送者使用私有密钥加密，而由接收者使用发送者的公开密钥解密。您可以使用接收者的公开密钥来对消息进行加密，然后接收者使用自己的私有密钥对消息进行解密。

  用来创建公开和私有密钥的算法更复杂，因此更不容易破译。但是，公开/私有密钥加密需要更多的计算、需要通过连接发送更多的数据，因而数据传输速度会明显降低。

键交换

减少计算开销同时加快事务处理速度，而又不牺牲安全性的方案是：综合使用对称密钥和公开/私有密钥加密，即所谓的 “键交换”。

对于大量数据，用对称密钥来加密原始消息。然后，发送者使用自己的私有密钥或接收者的公开密钥来加密对称密钥。加密消息和加密过的对称密钥都将发送给接收者。根据用于加密消息的密钥 （公开或私有），接收者使用相应的策略来解密对称密钥。一旦键被交换，接收者使用对称密钥来解密消息。

数字签名

数字签名用于篡改检测和不否认功能。数字签名通过数学算法创建，从文本消息生成唯一的、固定长度的数字字符串；其结果称作散列或消息摘要。

为了确保消息完整性，将使用签名者的私有密钥对消息摘要进行加密，然后将消息摘要和有关散列算法的信息一起发送给接收者。接收者用签名者的公开密钥对该消息进行解密。此过程还会重新生成原始消息摘要。如果该摘要匹配，则该消息完整且没被篡改。如果不匹配，则数据在转接过程中被更改，或者是被冒名者签名。

此外，数字签名提供**不否认**功能 — 发送者不能拒绝或否认他们发送了消息，因为他们的私有密钥对消息进行了加密。很显然，如果私有密钥产生危险 （被偷窃或解译），则对于 “不否认”功能，数字签名将变得毫无价值。

数字认证

数字认证类似于护照：一旦分配给您一个认证，权威部门会有您在系统中的所有标识（身份）信息。与护照类似，认证可用来检验一个实体（服务器、路由器及 Web 站点等）对另一个实体的标识。

Adaptive Server 使用两种类型的认证：

- **服务器认证** — 服务器认证对持有该认证的服务器进行鉴定。认证由受托第三方证书发放机构 (CA) 发布。 CA 验证拥有者的标识，将拥有者的公开密钥和其它鉴定信息嵌入到数字认证中。认证也包含发布 CA、验证所含数据的完整性，以及检验其使用状况的数字签名。
- **CA 认证**（也称作**受托根认证**）— 是在启动时服务器装载的受托 CA 的列表。当服务器作为客户端运行时，例如在远程过程调用（RPC）期间，服务器将使用 CA 认证。 Adaptive Server 在启动时装载 CA 受托根认证。为 RPC 连接到远程服务器后， Adaptive Server 将验证签署远程服务器认证的 CA 是否为列在其自己的 CA 受托根文件中的 “受托” CA。如果不是，连接将会失败。

认证在一段时期内有效， CA 可以因为多种原因撤消该认证，例如当出现安全问题时。如果一个认证在会话期间被撤消，则会话连接继续。以后的登录尝试将失败。同样，当认证到期后，登录尝试将失败。

这些机制的联合使用保护了在因特网上传输的数据免遭窃取和篡改。这些机制也保护用户免受其他用户欺骗，即一个实体假装成另一个实体（电子欺骗），或者一个人或组织为了特定目的而充当别的用户，其真实企图是获得私有信息 （误传）。

SSL 概述

SSL 是一个通过安全网络连接发送在线级 (wire-level) 或套接字级 (socket-level) 加密的数据的行业标准。

建立 SSL 连接之前，服务器和客户端交换一系列 I/O 往返，就安全性加密会话达成一致。该过程称为 “SSL 握手”。

SSL 握手

客户端请求一个连接时，在传输数据前，启用 SSL 的服务器提出认证以证实客户端的标识。“握手”主要由以下步骤组成：

- 客户端向服务器发送连接请求。该请求包括客户端所支持的 SSL（或传送层安全性， TLS）选项。
- 服务器返回其认证和所支持的密码成套程序列表，该列表包括 SSL/TLS 支持选项、键交换所用算法和数字签名。
- 当客户端和服务器就 CipherSuite 达成一致后，将建立一个安全、加密会话。

有关 **SSL 握手**和 SSL/TLS 协议的更详细信息，请参见 Internet Engineering Task Force Web 站点 （网址为 http://www.ietf.org）。

有关 Adaptive Server 支持的密码成套程序的列表，请参见第 239 页的 “密码成套程序”。

Adaptive Server 中的 SSL

Adaptive Server 的 SSL 实现提供了几种级别的安全性。

- 服务器鉴定自身 — 证实它是否是您所要联系的服务器，并在数据传输之前加密 SSL 会话。
- 一旦 SSL 会话建立后，要求连接的客户便可以通过此安全、加密的连接发送他的用户名和口令。
- 服务器认证时对数字签名进行比较，这样可以确定客户端所收到的数据在到达预期接收者之前是否被修改过。

在大多数平台上， Adaptive Server 使用 Certicom Corp. 提供的 SSL Plus(TM) library API。但是，对于 Windows Opteron X64， Adaptive Server 使用 OpenSSL 作为 SSL 提供程序。

SSL 过滤器

Adaptive Server 目录服务 （例如 *interfaces* 文件、 Windows 注册表或 LDAP 服务）定义了服务器地址和端口号，并且确定为客户端连接而强制执行的安全协议。 Adaptive Server 将 SSL 协议作为过滤器来执行，附加到目录服务的 master 行和 query 行。

Adaptive Server 在其上接受连接的地址和端口号是可配置的，因此您可以为单服务器启用多网络和安全协议。服务器连接属性由目录服务（如 LDAP）指定，或者由传统 Sybase *interfaces* 文件指定。请参见第 236 页的 “创建服务器目录条目”。

所有尝试到具有 **SSL 过滤器**的 *interfaces* 文件中 master 或 query 条目的连接都必须支持 SSL 协议。服务器可以配置为接受 SSL 连接，并且可拥有其它接受明码通信报文 （未加密的数据）或使用其它安全性机制的连接。

例如，UNIX 的 *interfaces* 文件，它支持基于 SSL 的连接和明码通信报文连接，类似以下内容：

```
SYBSRV1
master tcp ether myhostname myport1 ssl
query    tcp ether myhostname myport1 ssl
master tcp ether myhostname myport2
```

与使用 *interfaces* 文件 （Windows 上为 *sql.ini*）中的 SECMECH （安全性机制）行定义的其它安全性机制 （例如 Kerberos）相比， SSL 过滤器有所不同。

通过认证进行的鉴定

SSL 协议需要通过服务器认证的服务器鉴定来启用加密会话。同样，当 Adaptive Server 在 RPC 期间作为客户端运行时，必须有一个受托 CA 存储库，客户端连接可以访问该库以验证服务器认证。

服务器认证

每个 Adaptive Server 必须拥有自己的服务器认证文件，该文件在启动时装载。下面是认证文件的缺省位置，其中 *servername* 是启动过程中使用 -s 标志在命令行上指定或通过环境变量 *$DSLISTEN* 指定的 Adaptive Server 的名称。

- UNIX — *$SYBASE/$SYBASE_ASE/certificates/servername.crt*
- Windows — *%SYBASE%\%SYBASE_ASE%\certificates\servername.crt*

服务器认证文件由编码数据组成，包含服务器的认证和服务器认证的加密私有密钥。

或者，可以在使用 sp_ssladmin 时指定服务器认证文件的位置。

注释　若要成功进行客户端连接，认证中的公用名必须与 *interfaces* 文件中的 Adaptive Server 名称匹配。

CA 受托根认证

Adaptive Server 在启动时从受托根文件装载受托 CA 列表。受托根文件在格式上与认证文件类似，只是受托根文件包含 Adaptive Server 所知的 CA 认证。本地 Adaptive Server 可在以下文件中访问受托根文件，其中 *servername* 是服务器的名称：

- UNIX — *$SYBASE/$SYBASE_ASE/certificates/servername.txt*
- Windows — *%SYBASE%\%SYBASE_ASE\certificates\servername.txt*

仅当 Adaptive Server 作为客户端运行时 （例如 RPC 调用或组件集成服务 (CIS) 连接时），才会使用受托根文件。

系统安全员使用标准 ASCII 文本编辑器添加和删除 Adaptive Server 所要接受的 CA。

警告！ 使用 Adaptive Server 内的系统安全员角色 (sso_role) 来限制访问和执行安全性敏感的对象。

Adaptive Server 提供工具以产生认证请求和批准认证。请参见第 235 页的 “使用 Adaptive Server 工具请求和授权认证”。

连接类型

该节说明不同的客户端到服务器和服务器到服务器连接。

客户机登录到 Adaptive Server

Open Client 应用程序建立与 Adaptive Server 的套接字连接，类似于建立现有的客户端连接的方式。在传输任何用户数据之前，当客户端完成网络传送级连接调用且服务器端完成接受调用时，在套接口上将产生 “SSL 握手”。

服务器到服务器的远程过程调用

Adaptive Server 为 RPC （远程过程调用）建立的到另一服务器的套接字连接的方式，与现有 RPC 建立的连接方式相同。在传输任何用户数据之前，当网络传送级连接调用完成时，在套接口上将产生 “SSL 握手”。如果已经建立服务器到服务器套接字连接，那么现有套接字连接和安全性环境将被再次使用。

Adaptive Server 在 RPC 期间作为客户端运行时，它会在连接过程中请求远程服务器的认证。然后 Adaptive Server 验证签署远程服务器认证的 CA 是否受托；也就是说，验证该 CA 是否位于受托根文件中其自己受托 CA 的列表上。同时也验证服务器认证中的公用名是否与建立连接时使用的公用名匹配。

协同服务器和 SSL

可以使用协同服务器为故障切换配置 Adaptive Server。必须用相同的 SSL 和 RPC 配置参数来配置主服务器和辅助服务器。当连接进行故障切换或故障恢复时，连接将重新建立安全性会话。

Open Client 连接

组件集成服务、RepAgent、分布式事务管理和其它 Adaptive Server 中的模块，使用 Client-Library 来建立与服务器 （除了 Adaptive Server 外）的连接。远程服务器由其自身的认证体系来进行鉴定。远程服务器使用用户名和口令鉴定用于 RPC（远程过程调用）的 Adaptive Server 客户端连接。

启用 SSL

Adaptive Server 确定对于基于接口文件 （Windows 的 *sql.ini*）的端口将使用何种安全服务。

❖ **启用 SSL**

1　生成服务器的认证。

2　创建受托根文件。

3　使用 sp_configure 启用 SSL。从命令提示符中输入：

```
sp_configure "enable ssl", 1
```

- 1 — 在启动时启用 SSL 子系统，分配内存， SSL 在网络上对数据执行线级加密。
- 0 （缺省值）— 禁用 SSL。该值是缺省值。

4　将 SSL 过滤器添加到 *interfaces* 文件中。请参见第 236 页的 “创建服务器目录条目”。

5　使用 sp_ssladmin 将认证添加到认证文件中。请参见第 236 页的 “管理认证”。

6　关闭并重新启动 Adaptive Server。

注释　若要请求、授权和转换第三方认证，请参见 《实用程序指南》了解有关 certauth、 certreq 和 certpk12 工具的信息。

不同于其它安全服务，例如 Kerberos 和 NTLAN， SSL 既不依赖 Open Client/Open Server 配置文件 *libtcl.cfg* 的 “安全性”部分，也不依赖 *objectid.dat* 中的对象。

系统管理员在规划总物理内存时应考虑 SSL 使用的内存。 Adaptive Server 中的每个连接 （连接包含用户连接、远程服务器和网络监听器）大概需要 40K 才能进行 SSL 连接。内存保留和预分配在内存池中，当需要时由 Adaptive Server 和 SSL Plus libraries 在内部使用。

获得认证

系统安全员通过以下方式为 Adaptive Server 安装服务器认证和私有密钥：

- 使用与已经配置在客户环境下的现有公开密钥体系一起提供的第三方工具。
- 将 Adaptive Server 认证请求工具和受托第三方 CA 一起使用。

要获得认证，您必须从证书发放机构 (CA) 请求认证。 Adaptive Server 必须具有 SSL 证书才能使用 PEM 格式。但是，证书发放机构可以用非 PEM 格式发放证书。必须将证书转换为 PEM 格式。如果您从第三方请求证书，而且该证书采用 PKCS #12 格式，则应使用 certpk12 实用程序将证书转换为可被 Adaptive Server 理解的格式 （请参见 《实用程序指南》）。

为测试 Adaptive Server 认证请求工具，以及验证鉴定方法是否可以工作于您的服务器上， Adaptive Server 提供了一个供测试用的工具，该工具允许用户服务器作为 CA 运行以及向本机发布 CA 签署的认证。

创建用于 Adaptive Server 认证的主要步骤为：

1 生成公开和私有密钥对。

2 将私有密钥存储在安全的地方。

3 生成认证请求。

4 向 CA 发送认证请求。

5 CA 签署和返回认证后，把认证存储在文件中，然后对认证附加私有密钥。

6 把认证存储在 Adaptive Server 安装目录下。

请求认证的第三方工具

大多数第三方 PKI 供应商和一些浏览器包含生成认证和私有密钥的实用程序。通常，这些实用程序的形式为图形向导，它通过一系列问题提示您来定义认证的区分名和公用名。

遵照向导提供的指导来创建认证请求。一旦收到已签署的 PKCS #12 格式的认证，请使用 certpk12 生成认证文件和私有密钥文件。将两个文件并置到 *servername.crt* 文件 （其中 *servername* 是服务器名）中，然后将该文件放置在 *$SYBASE/$SYBASE_ASE* 下的 *certificates* 目录中。请参见 《实用程序指南》。

使用 Adaptive Server 工具请求和授权认证

Adaptive Server 提供两个用于请求和授权认证的工具。 certreq 生成公开和私有密钥对以及认证请求。 certauth 将服务器认证请求转换为 CA 签署的认证。

警告！ 仅将 certauth 作为测试之用。Sybase 建议您使用商业 CA 的服务，因为它对根认证的完整性提供了保护，而且由普遍公认的 CA 签署的认证容易移植到使用鉴定的客户端认证环境下。

准备服务器受托根认证的过程包含五个步骤。执行前两个步骤来创建测试受托根认证，以便您可以验证用户是否能够创建服务器认证。获得测试 CA 认证 （受托根认证）后，重复执行步骤 3 到步骤 5 以签署服务器认证。

1　使用 certreq 请求认证。

2　使用 certauth 将认证请求转换为 CA 自签的认证 （受托根认证）。

3　使用 certreq 请求服务器认证和私有密钥。

4　使用 certauth 将认证请求转换为 CA 签署的服务器认证。

5　将私有密钥文本附加到服务器认证，并将该认证存储在服务器安装目录中。

注释　Adaptive Server 在 *$SYBASE/$SYBASE_OCS/bin* 目录中包括 openssl 开放源代码实用程序。使用 openssl 可完成由 certreq、certauth 和 certpk12 实现的所有认证管理任务。 Sybase 包含此二进制程序是为了方便起见，对使用该二进制程序所引致的任何问题概不负责。有关详细信息，请参见 www.openssl.org。

有关 Sybase 实用程序，即用于请求、授权和转换第三方认证的 certauth、certreq 和 certpk12 的信息，请参见 《实用程序指南》。

注释　certauth 和 certreq 依赖于 RSA 和 DSA 算法。这些工具仅用于 crypto 模块，该模块使用 RSA 和 DSA 算法来构造认证请求。

创建服务器目录条目

Adaptive Server 接受客户端登录和服务器到服务器的 RPC。 Adaptive Server 接受连接的地址和端口号是可配置的，因此可以指定多网络、不同的协议和替代端口。

在 *interfaces* 文件中，SSL 被指定为 master 和 query 行上的过滤器，而诸如 Kerberos 等安全性机制则使用 SECMECH 行标识。以下示例显示在 UNIX 环境下，使用 SSL 的 Adaptive Server 的，基于 TLI 的条目：

Windows 上使用 SSL 和 Kerberos 安全性机制的 Adaptive Server 的条目可能类似于：

```
[SYBSRV2]
    query=nlwnsck, 18.52.86.120,2748,ssl
    master=nlwnsck 18.52.86.120,2748,ssl
    master=nlwnsck 18.52.86.120,2749
    secmech=1.3.6.1.4.897.4.6.6
```

示例中 SYBSRV2 的 SECMECH 行包含对象标识符 (OID)，它表示安全性机制 Kerberos。 OID （对象标识符）的值在以下文件中定义：

- UNIX — *$SYBASE/$SYBASE_OCS/config/objectid.dat*
- Windows — *%SYBASE%\%SYBASE_OCS\ini\objectid.dat*

这些示例中， SSL 安全服务在端口号 2748(0x0abc) 上指定。

注释 同时使用 SSL 和 SECMECH 安全性机制的目的是便于从 SECMECH 迁移到 SSL 安全性。

管理认证

要管理 Adaptive Server 中的 SSL 和认证，请使用 sp_ssladmin。 sso_role 是执行存储过程所必需的。

sp_ssladmin 用于：

- 增加本地服务器认证。可以增加认证和指定用来加密私有密钥的口令，或者启动期间需要在命令行上输出口令。
- 删除本地服务器认证。
- 列出服务器认证。

sp_ssladmin 的语法是：

```
sp_ssladmin {[addcert, certificate_path [, password|NULL]]
    [dropcert, certificate_path]
    [lscert]
    [help]}
    [lsciphers]
    [setciphers, {"FIPS" | "Strong" | "Weak" | "All"
    | quoted_list_of_ciphersuites}]
```

例如：

```
sp_ssladmin addcert, "/sybase/ASE-12_5/certificates/Server1.crt",
      "mypassword"
```

此示例为本地服务器添加一个条目，即 *Server1.crt*，该条目位于绝对路径 */sybase/ASE-12_5/certificates*（Windows 上为 *x:\sybase\ASE-12_5\certificates*）下的认证文件中。私有密钥用口令 “*mypassword*” 进行加密，加密口令应为在创建私有密钥时指定的口令。

接受认证之前， sp_ssladmin 验证：

- 可以使用提供的口令对私有密钥解密 （被指定为 NULL 时除外）。
- 认证中的私有密钥和公开密钥匹配。
- 从根 CA 到服务器认证的认证链有效。
- 认证中的公用名和 *interfaces* 文件中的公用名匹配。

如果公用名不匹配， sp_ssladmin 发出警告。如果其它标准失败，则不会将认证添加到认证文件中。

警告！ Adaptive Server 将口令限制为 64 个字符。此外，某些平台在创建服务器认证时限制有效口令的长度。在以下限制范围内选择口令：

- Sun Solaris — 32 位和 64 位平台， 256 个字符。
- Linux — 128 个字符。
- IBM — 32 位和 64 位平台， 32 个字符。
- HP — 32 位和 64 位平台， 8 个字符。
- Windows — 256 个字符。

使用 NULL 作为口令的目的是在 SSL 加密会话开始之前的初始配置 SSL 期间保护口令。因为还没有配置 SSL，所以连接过程中口令没有加密。第一次登录时，通过将口令指定为 NULL 可以避免上述情况。

口令为 NULL 时，您必须使用 -y 标志启动 dataserver，该标志在命令行提示系统管理员输入私有密钥口令。

在建立了 SSL 连接并且重新启动了 Adaptive Server 后，再次使用 sp_ssladmin，这一次使用实际口令。这时， Adaptive Server 将加密并存储口令。以后从命令行启动 Adaptive Server 都要使用加密的口令；启动期间不必在命令行指定口令。

第一次登录时使用 NULL 口令的另一种替代方法是，通过 isql 避免远程连接到 Adaptive Server。可以指定 “localhost”作为 *interfaces* 文件（Windows 上为 *sql.ini*）中的 *hostname*，以防止客户端以远程方式连接。只能建立本地连接，并且口令绝不会通过网络连接传送。

注释 Adaptive Server 在其网络内存池中具有足够的内存，从而允许 sp_ssladmin addcert 使用其缺省的内存分配来设置认证和私有密钥口令。但是，如果另一网络内存消耗程序已被分配了缺省网络内存，则 sp_ssladmin 可能会失败并向客户端显示以下错误：

```
Msg 12823, Level 16, State 1:
Server 'servername', Procedure 'sp_ssladmin', Line 72:
Command 'addcert' failed to add certificate path
/work/REL125/ASE-12_5/certificates/servername.crt,
system error:ErrMemory.
(return status = 1)
```

或者可能在错误日志中显示以下消息：

```
 ... ssl_alloc:Cannot allocate using
ubfalloc(rnetmempool, 131072)
```

一种解决方法是：您可以增加 additional network memory 配置参数。Adaptive Server 需要大约 500K 字节的内存才能使 sp_ssladmin addcert 成功，因此将额外的网络内存增大到该数量可能会使其成功。网络内存池在需要时会再次使用该内存，或者您可以在 sp_ssladmin 成功完成后将 additional network memory 返回为它以前的值。

性能

建立安全会话需要额外的开销，因为数据加密后数据的大小将增加，而且需要额外的计算加密或解密信息。 SSL 的额外内存要求将使用于网络吞吐量或建立连接的开销增加 50%-60%。每个用户连接必须有大约 40K 或更多的内存。

密码成套程序

SSL 握手期间，客户端和服务器通过密码成套程序对安全性协议达成一致。**密码成套程序**是启用 SSL 的应用程序所使用的密钥交换算法、散列方法和加密方法的优先列表。有关密码成套程序的详细说明，请访问 Internet Engineering Task Force (IETF) 组织 （网址 http://www.ietf.org/rfc/rfc2246.txt）。

缺省情况下，客户端和服务器均支持的最强大密码成套程序是用于基于 SSL 会话的密码成套程序。

Adaptive Server 支持对 SSL Plus library API 和密码引擎 Security Builder™ （均由 Certicom Corp. 提供）都可用的密码成套程序。

注释　列出的密码成套程序符合传送层规范 (TLS)。TLS 是 SSL 3.0 的增强版本，并且是 SSL 3.0 版密码成套程序的一个别名。

@@ssl_ciphersuite

通过 Transact-SQL 全局变量 *@@ssl_ciphersuite*，用户可以了解 SSL 握手选择了哪个密码成套程序，并验证建立的连接是否为 SSL 连接。

Adaptive Server 在 SSL 握手完成时设置 *@@ssl_ciphersuite*。其值可以是 NULL （表示非 SSL 连接），或包含 SSL 握手选择的密码成套程序的名称的字符串。

例如，使用 SSL 协议的 isql 连接会显示为其选择的密码成套程序。

```
1>  select @@ssl_ciphersuite
2>   go
```

输出：

```
------------------------------
TLS_RSA_WITH_AES_128_CBC_SHA

(1 row affected)
```

设置 SSL 密码成套程序优先选项

在 Adaptive Server 中，sp_ssladmin 具有两个命令选项用于显示和设置密码成套程序优先选项：lsciphers 和 setciphers。使用这些选项，可以限制 Adaptive Server 使用的密码成套程序集，这样，系统安全员便能够控制客户端到服务器的连接或 Adaptive Server 的出站连接可使用的加密算法的种类。在 Adaptive Server 中，缺省根据内部定义的密码成套程序优先选项集来使用 SSL 密码成套程序，这一点与早期版本中相同。

若要显示密码成套程序优先选项集的值，请输入：

```
sp_ssladmin lsciphers
```

若要设置特定的密码成套程序优先选项，请输入：

```
sp_ssladmin setciphers, {"FIPS" | "Strong" | "Weak" |
"All" | quoted_list_of_ciphersuites }
```

其中：

- "FIPS" — 是符合 FIPS 要求的加密算法、散列算法和密钥交换算法的集合。此列表包括的算法有 AES、 3DES、 DES 和 SHA1。
- "Strong" — 是所用密钥长度大于 64 位的加密算法的集合。
- "Weak" — 是属于所有受支持的密码成套程序的集合但又不属于 strong 集合的加密算法的集合。
- "All" — 是缺省密码成套程序的集合。
- quoted_list_of_ciphersuites — 以逗号分隔的列表形式指定密码成套程序集，列表中各项按优先顺序排列。该列表首尾用引号 (“) 括起。此带引号的列表可以包含任何预定义集合以及单个密码成套程序名称。未知的密码成套程序名称将导致报告错误，但对优先选项无影响。

第 241 页的表 7-1 中包含预定义集合的详细内容。

sp_ssladmin setciphers 将密码成套程序优先选项设置为给定的有序列表。这样可将可用的 SSL 密码成套程序限制为由 "FIPS"、 "Strong"、 "Weak"、 "All" 组成的指定集合或带引号的密码成套程序列表。此设置会在下一个监听器启动时生效，并且需要您重新启动 Adaptive Server 以确保所有监听器都使用新设置。

您可以显示已经使用 sp_ssladmin lsciphers 设置的任何密码成套程序优先选项。如果没有设置任何优先选项， sp_ssladmin lsciphers 将返回 0 行，表示未设置任何优先选项，并且 Adaptive Server 将使用其缺省 （内部） 优先选项。

表 7-1：Adaptive Server 中的预定义密码成套程序

集名称	集中包含的密码成套程序的名称
FIPS	TLS_RSA_WITH_AES_256_CBC_SHA
	TLS_RSA_WITH_AES_128_CBC_SHA
	TLS_RSA_WITH_3DES_EDE_CBC_SHA
	TLS_DHE_DSS_WITH_3DES_EDE_CBC_SHA
	TLS_DHE_RSA_WITH_3DES_EDE_CBC_SHA
	TLS_RSA_WITH_DES_CBC_SHA
	TLS_DHE_DSS_WITH_DES_CBC_SHA
	TLS_DHE_RSA_WITH_DES_CBC_SHA
	TLS_RSA_EXPORT1024_WITH_DES_CBC_SHA
	TLS_DHE_DSS_EXPORT1024_WITH_DES_CBC_SHA
Strong	TLS_RSA_WITH_AES_256_CBC_SHA
	TLS_RSA_WITH_AES_128_CBC_SHA
	TLS_RSA_WITH_3DES_EDE_CBC_SHA
	TLS_RSA_WITH_RC4_128_SHA
	TLS_RSA_WITH_RC4_128_MD5
	TLS_DHE_DSS_WITH_3DES_EDE_CBC_SHA
	TLS_DHE_DSS_WITH_RC4_128_SHA
	TLS_DHE_RSA_WITH_3DES_EDE_CBC_SHA
Weak	TLS_RSA_WITH_DES_CBC_SHA
	TLS_DHE_DSS_WITH_DES_CBC_SHA
	TLS_DHE_RSA_WITH_DES_CBC_SHA
	TLS_RSA_EXPORT1024_WITH_DES_CBC_SHA
	TLS_RSA_EXPORT1024_WITH_RC4_56_SHA
	TLS_DHE_DSS_EXPORT1024_WITH_RC4_56_SHA
	TLS_DHE_DSS_EXPORT1024_WITH_DES_CBC_SHA
	TLS_RSA_EXPORT_WITH_RC4_40_MD5
	TLS_RSA_EXPORT_WITH_DES40_CBC_SHA
	TLS_DHE_DSS_EXPORT_WITH_DES40_CBC_SHA
	TLS_DHE_RSA_EXPORT_WITH_DES40_CBC_SHA

集名称	集中包含的密码成套程序的名称
All	TLS_RSA_WITH_AES_256_CBC_SHA
	TLS_RSA_WITH_AES_128_CBC_SHA
	TLS_RSA_WITH_3DES_EDE_CBC_SHA
	TLS_RSA_WITH_RC4_128_SHA
	TLS_RSA_WITH_RC4_128_MD5
	TLS_DHE_DSS_WITH_3DES_EDE_CBC_SHA
	TLS_DHE_DSS_WITH_RC4_128_SHA
	TLS_DHE_RSA_WITH_3DES_EDE_CBC_SHA
	TLS_RSA_WITH_DES_CBC_SHA
	TLS_DHE_DSS_WITH_DES_CBC_SHA
	TLS_DHE_RSA_WITH_DES_CBC_SHA
	TLS_RSA_EXPORT1024_WITH_DES_CBC_SHA
	TLS_RSA_EXPORT1024_WITH_RC4_56_SHA
	TLS_DHE_DSS_EXPORT1024_WITH_RC4_56_SHA
	TLS_DHE_DSS_EXPORT1024_WITH_DES_CBC_SHA
	TLS_RSA_EXPORT_WITH_RC4_40_MD5
	TLS_RSA_EXPORT_WITH_DES40_CBC_SHA
	TLS_DHE_DSS_EXPORT_WITH_DES40_CBC_SHA
	TLS_DHE_RSA_EXPORT_WITH_DES40_CBC_SHA

表 7-2 介绍了 Adaptive Server 15.0 和更高版本不再支持的密码成套程序。15.0. 如果尝试使用任何删除的密码成套程序，将导致 SSLHandshake 失败，并且无法连接到 Adaptive Server。

表 7-2：删除的密码成套程序

集名称	集中删除的密码成套程序的名称
FIPS	TLS_RSA_EXPORT1024_WITH_DES_CBC_SHA
Strong	未删除
Weak	TLS_RSA_EXPORT1024_WITH_RC4_56_SHA
	TLS_DHE_DSS_EXPORT_WITH_DES40_CBC_SHA
	TLS_DHE_RSA_EXPORT_WITH_DES40_CBC_SHA
其它	TLS_DH_anon_EXPORT_WITH_DES40_CBC_SHA
	TLS_DH_anon_EXPORT_WITH_RC4_40_MD5
	TLS_DH_anon_WITH_3DES_EDE_CBC_SHA
	TLS_DH_anon_WITH_DES_CBC_SHA
	TLS_DH_anon_WITH_RC4_128_MD5
	TLS_RSA_WITH_NULL_MD5
	TLS_RSA_WITH_NULL_SHA

sp_ssladmin 示例

最初启动时，在设置任何密码成套程序优先选项前， sp_ssladmin lscipher 不显示任何优先选项。

```
1> sp_ssladmin lscipher
2> go
```

输出：

```
 Cipher Suite Name   Preference
-----------------    ----------
(0 rows affected)
(return status = 0)
```

以下示例指定使用 FIPS 算法的密码成套程序集。

```
1> sp_ssladmin setcipher, 'FIPS'
The following cipher suites and order of preference are set for SSL connections:

Cipher Suite Name                                                Preference
---------------------------------------------------------------- -----------
TLS_RSA_WITH_AES_256_CBC_SHA                                               1
TLS_RSA_WITH_AES_128_CBC_SHA                                               2
TLS_RSA_WITH_3DES_EDE_CBC_SHA                                              3
TLS_DHE_DSS_WITH_3DES_EDE_CBC_SHA                                          4
TLS_DHE_RSA_WITH_3DES_EDE_CBC_SHA                                          5
TLS_RSA_WITH_DES_CBC_SHA                                                   6
TLS_DHE_DSS_WITH_DES_CBC_SHA                                               7
TLS_DHE_RSA_WITH_DES_CBC_SHA                                               8
```

```
TLS_RSA_EXPORT1024_WITH_DES_CBC_SHA                                    9
TLS_DHE_DSS_EXPORT1024_WITH_DES_CBC_SHA                                10
```

sp_ssladmin 输出的优先选项为 0 （零）表示 Adaptive Server 未使用密码成套程序。如果输出的是其它非零值，则该数值表示 Adaptive Server 在 SSL 握手过程中使用该算法的优先顺序。 SSL 握手的客户端选择这些密码成套程序中与其接受的密码成套程序的列表匹配的一个密码成套程序。

此示例使用带引号的密码成套程序列表来设置 Adaptive Server 中的优先选项：

```
1> sp_ssladmin setcipher, 'TLS_RSA_WITH_AES_128_CBC_SHA,
TLS_RSA_WITH_AES_256_CBC_SHA'
2> go
The following cipher suites and order of preference are set for SSL connections:
Cipher Suite Name                                                  Preference
------------------------------------------------------------------ -----------
TLS_RSA_WITH_AES_128_CBC_SHA                                                1
TLS_RSA_WITH_AES_256_CBC_SHA                                                2
```

其它注意事项

升级到 Adaptive Server 12.5.3 版及更高版本时，密码成套程序优先选项为服务器缺省值，并且 sp_ssladmin 选项 lscipher 不显示任何优先选项。服务器使用其缺省优先选项，这些优先选项由 "All" 定义。系统安全员应考虑在其节点使用的安全性策略以及可用的 SSL 密码成套程序，以决定是否限制密码成套程序，以及哪些密码成套程序适合于安全性策略。

如果从 Adaptive Server 12.5.3 版及更高版本升级，并且设置了密码成套程序优先选项，则这些优先选项会在升级后保留。升级完成后，查看具有当前安全性策略的服务器的密码成套程序优先选项，以及在表 7-1 中找到的受支持和不受支持的密码成套程序列表。忽略任何不受支持的密码成套程序。

如果已设置 SSL 密码成套程序优先选项，并且需要从服务器上删除所有优先选项并使用缺省优先选项，请使用下列命令，将优先选项从其在系统目录中的存储位置中删除：

```
1> sp_configure 'allow updates to system tables', 1
2> go

1>  delete from master..sysattributes where class=24
2>   go

1> sp_configure 'allow updates to system tables', 0
2> go
```

只有系统安全员或系统管理员才能执行这些命令。

使用 SSL 指定公用名

目录服务条目中指定的服务器名可与 SSL 服务器认证用于执行 SSL 握手的公用名不同。这样，您将能够为 SSL 认证公用名使用完全限定域名（例如， *server1.bigcompany.com*）。

若要将公用名添加到 interfaces 文件中，请使用：

```
ase1
   master tcp ether host_name port_number ssl="CN='common_name'"
   query tcp ether host_name port_number ssl="CN='common_name'"
```

当客户端使用 SSL 连接到同样使用 SSL 的 Adaptive Server 时，将在 *interfaces* 文件中的端口号后面放置 SSL 过滤器。目录服务包括您使用 dsedit 或文本编辑器添加的公用名。

使用 *sp_listener* 指定公用名

sp_listener 包括 CN=*common_name* 参数，从而使您能够为 SSL 证书指定公用名。语法为：

```
sp_listener 'command','[protocol:]machine_name:port_number:
"CN=common_name"', 'engine_number'
```

其中，只有当将 ssltcp 指定为协议时，才使用 CN=*common_name*。将依据 SSL 认证中的 *common_name* 对您在此处指定的 *common_name* 进行验证。如果未包括 CN=*common_name*，Adaptive Server 将使用 *server_name*，依据 SSL 认证中的公用名进行验证。如果在证书中包括完全限定域名，该名称必须与 CN=*common_name* 匹配。

属性名 “CN”不区分大小写 （可为 “CN”、“cn”或 “Cn”），但公用名的属性值区分大小写。

例如，若要指定公用名 `ase1.big server 1.com`：

```
sp_listener 'start','ssltcp:blade1:17251:"CN=ase1.big server 1.com"','0'
```

有关 sp_listener 的详细信息，请参见 《参考手册：过程》。

存储过程 *sp_addserver* 已更改

filter 参数得以增强，可以指定公用名。请参见 《参考手册：过程》。

Kerberos 保密性

您还可以使用 Adaptive Server 确保所有消息的保密性。若要要求对进出 Adaptive Server 的所有消息进行加密，请将 msg confidentiality reqd 配置参数设置为 1。如果此参数为 0 （缺省值），则不要求消息的保密性，但客户端仍可以建立消息保密性。

例如，若要求所有消息被加密，需执行：

```
sp_configure "msg confidentiality reqd", 1
```

有关使用 Message Confidentiality with Kerberos 和其它受支持的安全服务的详细信息，请参见第 88 页的 “管理基于网络的安全性”。

转储和装载数据库时使用口令保护

可以使用 dump database 命令的 password 参数保护数据库转储，使其免于未经授权的装载。如果在进行数据库转储时包括了 password 参数，则在装载该数据库时，也必须提供相同的口令。

带口令保护的 dump database 和 load database 命令的部分语法为：

dump database *database_name* to *file_name* [with passwd = *password*]

load database *database_name* from *file_name* [with passwd = *password*]

其中：

- *database_name* — 是进行转储或装载的数据库的名称。
- *file_name* — 是转储文件的名称。
- *password* — 是您提供的口令，用来保护转储文件免于被未经授权的用户使用。

口令长度必须介于 6 到 30 个字符之间。如果提供的口令少于 6 个字符或多于 30 个字符，Adaptive Server 将发出错误消息。如果在尝试装载数据库时发出了不正确的口令， Adaptive Server 将发出错误消息，并且命令将失败。

例如，以下命令使用口令 “bluesky” 保护 pubs2 数据库的数据库转储：

```
dump database pubs2 to "/Syb_backup/mydb.db" with passwd = "bluesky"
```

装载数据库转储时必须使用相同的口令：

```
load database pubs2 from "/Syb_backup/mydb.db" with passwd = "bluesky"
```

口令与 Adaptive Server 的早期版本

只能在 Adaptive Server 12.5.2 版和更高版本中使用带口令保护的 dump 和 load 命令。如果对 Adaptive Server 12.5.2 版的转储使用口令参数，则在尝试将此转储装载到 Adaptive Server 的早期版本上时，装载会失败。

口令和字符集

只能将转储装载到另一台具有相同字符集的服务器上。例如，如果试图将使用 ASCII 字符集的服务器中的转储装载到一台使用非 ASCII 字符集的服务器上，由于 ASCII 口令的值不同于非 ASCII 口令的值，装载会失败。

用户输入的口令会转换为 Adaptive Server 的本地字符集。由于 ASCII 字符在字符集间通常表示相同的值，因此，如果某个用户的口令使用的是 ASCII 字符集，则 dump 和 load 的口令在所有字符集中都可以被识别。

Adaptive Server 15.0.2 版及更高版本允许您存储可移植口令。请参见第 37 页的 “口令的字符集注意事项”。

第 8 章

审计

本章介绍如何为安装设置审计。

Adaptive Server 中的审计简介

一个安全系统的基本要素是责任。确保责任的一种方法是审计系统中的事件。 Adaptive Server 中发生的许多事件都可以记录。

审计是数据库管理系统中安全性的重要组成部分。可使用审计追踪来检测系统渗透和资源误用情况。通过检查审计追踪，系统安全员可以对数据库中对象的访问模式进行检测，并可监视特定用户的活动。可以跟踪特定用户的审计记录，这对于不正确使用系统的用户可起到威慑作用。

每个审计记录都可以记录事件的性质、日期和时间、对它负责的用户以及事件是否成功。可以审计的事件包括登录和注销、服务器的启动、数据访问命令的使用、访问特定对象的尝试以及特定用户的操作。**审计追踪**（即审计记录的日志）允许系统安全员重建系统上发生的事件并评估其影响。

只有系统安全员才能启动和停止审计、设置审计选项以及处理审计数据。作为系统安全员，您可以为诸如以下事件建立审计：

- 全服务器范围内的与安全性相关的事件
- 创建、删除和修改数据库对象
- 特定用户执行的所有操作或具有特定活动角色的用户执行的所有操作

- 授予或撤消数据库访问权限
- 导入或导出数据
- 登录和注销

将 Adaptive Server 与操作系统的审计记录相关联

链接 Adaptive Server 审计记录和操作系统记录的最简便方法是使 Adaptive Server 的登录名与操作系统的登录名相同。

或者，系统安全员可将用户的操作系统登录名映射为其 Adaptive Server 登录名。然而，这种方法要求进行即时维护，因为新用户的登录名必须手动记录。

审计系统

审计系统包括：

- sybsecurity 数据库，包括全局审计选项和审计追踪
- 内存中的审计队列；在审计记录被写入到审计追踪中之前，将其发送到队列中
- 用于管理审计的配置参数
- 用于管理审计的系统过程

sybsecurity 数据库

sybsecurity 数据库是在审计安装过程中创建的。除 model 数据库中的所有系统表外，它还包括用来跟踪整个服务器范围内审计选项的 sysauditoptions 系统表和用于审计追踪的系统表。

sysauditoptions 包含全局审计选项的当前设置，例如是否为磁盘命令、远程过程调用、用户定义的即席审计记录或所有安全性相关事件启用了审计。这些选项会影响整个 Adaptive Server。

审计追踪

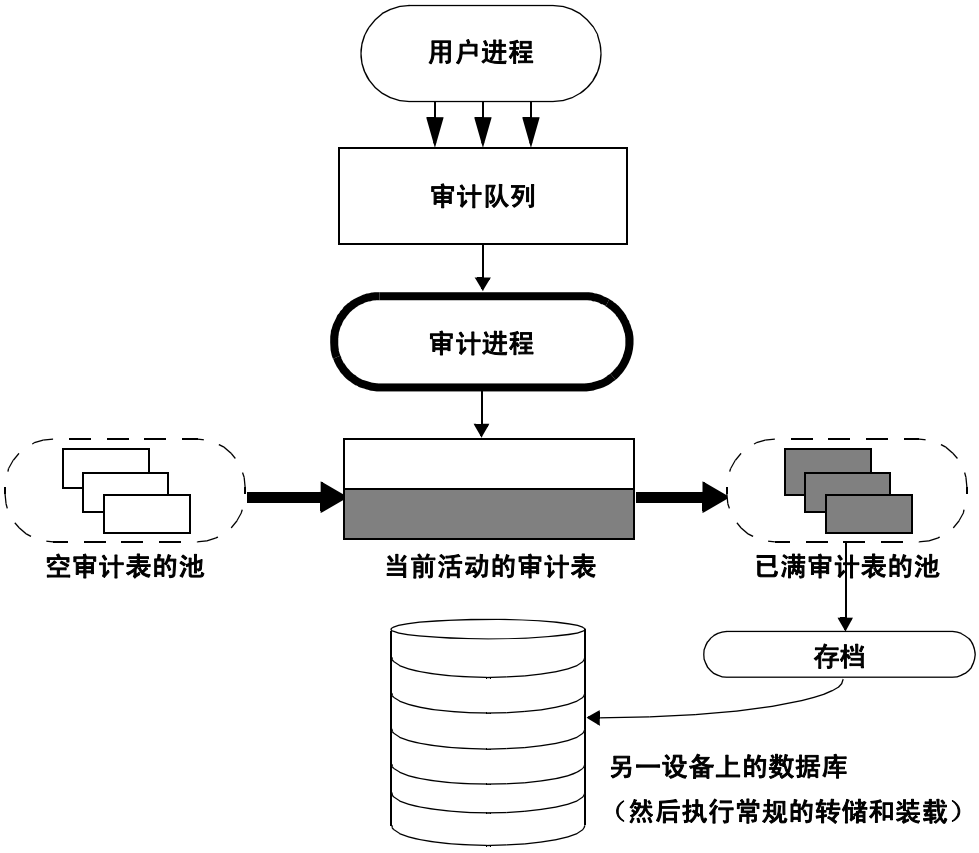
Adaptive Server 在名为 sysaudits_01 到 sysaudits_08 的系统表中存储审计追踪。安装审计时，需要确定要安装的审计表的数量。例如，如果选择使用两个审计表，其名称分别为 sysaudits_01 和 sysaudits_02。在任何给定时间，只有一个审计表是当前审计表。 Adaptive Server 将所有审计数据都写入当前审计表中。系统安全员可以使用 sp_configure 来设置（或更改）哪个审计表为当前表。

Sybase 建议使用两个或两个以上的审计表，且每个表都位于单独的审计设备上。这样就可以建立一个稳定运行的审计过程来存档和处理审计表，而不会丢失审计记录并且不需要手动干预。

警告！ Sybase 强烈反对在生产系统中仅使用一个审计表。如果仅使用一个审计表，会有可能丢失审计记录。如果由于系统资源有限而必须仅使用一个审计表，则请参见第 265 页的 “单表审计”以获得有关说明。

图 8-1 显示了审计进程如何处理多个审计表。

图 8-1：使用多个审计表进行审计



审计系统将来自内存中的审计队列的审计记录写入到当前审计表中。当前审计表将近充满时，一个阈值过程可自动将该表存档到另一数据库中。使用 dump 和 load 命令，可以对存档数据库进行备份和恢复。使用存档数据库访问权限以只读方式访问备份中的已存档审计表。请参见 《系统管理指南：第二卷》中的第 14 章XXXXXXXX “存档数据库访问”。有关管理审计追踪的详细信息，请参见第 258 页的 “设置审计追踪管理”。

审计队列

当被审计的事件发生时，首先向内存中的审计队列增加一条审计记录。该记录一直保留在内存中，直到被审计进程写入审计追踪。用户可使用 sp_configure 的 audit queue size 参数配置审计队列的大小。

配置审计队列的大小之前，应综合考虑在系统崩溃时丢失队列中的记录的风险以及队列充满时性能降低的风险。只要审计记录在队列中，如果系统崩溃，记录就有可能丢失。然而，如果队列频繁充满，则整个系统的性能都会受影响。如果在用户进程尝试生成一条审计记录时审计队列已满，则该进程将休眠，直至队列中有可用空间为止。

注释　由于不会将审计记录直接写入审计追踪，因此不能期望审计记录会立即被存储到当前审计表中。

审计配置参数

可以使用下面这些配置参数来管理审计进程：

- auditing 启用或禁用对整个 Adaptive Server 的审计。此参数在执行 sp_configure 后立即生效。只有在启用此参数后，审计才会发生。
- audit queue size 确定审计队列的大小。由于此参数会影响内存分配，所以需重新启动 Adaptive Server 后才会生效。
- suspend audit when device full 控制在审计设备充满时审计进程的行为。此参数在执行 sp_configure 后立即生效。
- current audit table 设置当前审计表。此参数在执行 sp_configure 后立即生效。

用于审计的系统过程

可以使用下面这些系统过程来管理审计进程：

- sp_audit 启用和禁用审计选项。这是建立待审计事件所需的唯一系统过程。
- sp_displayaudit 显示活动的审计选项。
- sp_addauditrecord 将用户定义的审计记录（注释）添加到审计追踪中。只有在系统安全员使用 sp_audit 启用即席审计后，用户才能添加这些记录。

安装和设置审计

表 8-1：审计的常规步骤

操作和说明	请参见
1. 安装审计 — s 设置审计表的数量，并为审计追踪和 sybsecurity 数据库中的 syslogs 事务日志指派设备。	第 254 页的 “安装审计系统”以及 Adaptive Server 安装和配置文档
2. 设置审计追踪管理 — 编写和建立阈值过程，当前审计表将近充满时，由该过程接管。此过程将自动切换到新的审计表，并存档当前表的内容。 此外，该步骤还包括设置 audit queue size 和 suspend audit when device full 配置参数。	第 258 页的 “设置审计追踪管理” 有关单个表审计的说明，请参见第 265 页的 “单表审计”
3. 设置 sybsecurity 数据库中的事务日志管理 — 确定如何处理 sybsecurity 数据库中的 syslogs 事务日志；如何设置 trunc log on chkpt 数据库选项；以及在 trunc log on chkpt 设置为禁用的情况下如何为 syslogs 建立一个最后机会阈值过程。	第 264 页的 “设置事务日志管理”
4. 设置审计选项 — 使用 sp_audit 建立要审计的事件。	第 269 页的 “设置全局审计选项”
5. 启用审计 — 使用带有 auditing 配置参数的 sp_configure。Adaptive Server 开始将审计记录写入当前审计表。	第 265 页的 “启用和禁用审计”
6. 重新启动审计 — 使用 sp_audit restart 重新启动审计 （如果审计失败）。	第 268 页的 “重新启动审计”

安装审计系统

通常使用 Sybase 安装程序 auditinit 来安装审计系统。或者，也可在不使用 auditinit 的情况下安装审计。有关详细信息，请参见第 255 页的 “使用 installsecurity 安装审计”。所用平台的 Adaptive Server 安装和配置文档中论述了安装和 auditinit。

安装审计时，可以确定要用于审计追踪的系统表数量，用于每个审计系统表的设备以及用于 syslogs 事务日志的设备。

用于审计追踪的表和设备

最多可以指定八个系统表 （sysaudits_01 到 sysaudits_08）。应计划至少使用两个表用于审计追踪。将每个表分别放在自己的设备上，不要放在主设备上。这样就可以使用一个阈值过程，在当前审计表快填满前自动存档当前审计表，然后为随后的审计记录切换到新的空审计表。

用于 *syslogs* 事务日志表的设备

安装审计时，必须为由 syslogs 系统表所组成的事务日志指定一个单独的设备。 syslogs 表在每个数据库中都有，包含了数据库中执行的事务的日志。

使用 *installsecurity* 安装审计

$SYBASE/ASE-15_0/scripts 目录包含一个用于安装审计的脚本 *installsecurity*。

注释　此示例假定服务器使用 2K 的逻辑页大小。

使用 *installsecurity* 安装审计：

1　使用 disk init 和 create database 命令创建审计设备和审计数据库。例如：

```
disk init name = "auditdev",
    physname = "/dev/dsk/c2d0s4",
    size = "10M"
disk init name = "auditlogdev",
    physname = "/dev/dsk/c2d0s5",
    size = "2M"
create database sybsecurity on auditdev
    log on auditlogdev
```

2　使用 *isql* 执行 installsecurity 脚本：

```
cd $SYBASE/ASE-12_5/scripts
setenv DSQUERY server_name
isql -Usa -Ppassword -Sserver_name < installsecurity
```

3　关闭并重新启动 Adaptive Server。

当您完成这些步骤后， sybsecurity 数据库已在自己的段中创建了一个审计表 (sysaudits_01)。这时可以启用审计，但应使用 sp_addaudittable 添加更多的审计表。有关 disk init、 create database 和 sp_addaudittable 的信息，请参见 《参考手册：过程》。

将审计数据库移动到多个设备

将 sybsecurity 数据库放在其自己的设备上，与 master 数据库分开。如果有多个审计表，将每个表放在各自的设备上。把每个表放在指向独立设备的独立的段上也会有所帮助。如果当前您将 sybsecurity 与 master 保存在同一设备上，或想将 sybsecurity 移动到另一设备上，则使用下面描述的过程之一来完成。移动数据库时，可以指定是否保存现有的全局审计设置。

移动 *sybsecurity* 而不保存全局审计设置

注释 这些步骤包括删除 sybsecurity 数据库，该步骤将销毁所有审计记录以及 sybsecurity 中以前记录的全局审计设置。在删除 sybsecurity 数据库之前，请确保使用备份或按照第 259 页的 “存档审计表”中的说明将现有记录存档，以避免丢失保留在 sybsecurity 表中的任何历史数据。

若要移动 sybsecurity 数据库而不保存全局审计设置：

1 执行以下命令，从 syslogins 系统表中删除所有与登录相关的信息：

```
sp_audit "all","all","all","off"
```

2 删除 sybsecurity 数据库。

3 使用下面描述的两个安装过程之一重新安装 sybsecurity：

- 针对所用平台的配置文档，或
- 第 255 页的 “使用 installsecurity 安装审计”

4 安装过程中，将 sybsecurity 数据库放置在独立于主设备的一个或多个设备上。

移动 *sybsecurity*，然后保存全局审计设置

❖ **移动 *sybsecurity* 数据库并保存全局审计设置**

1 转储 sybsecurity 数据库：

```
dump database sybsecurity to "/remote/sec_file"
```

2 删除 sybsecurity 数据库：

```
drop database sybsecurity
```

3　初始化第一个要在其中放置 sybsecurity 数据库的设备：

```
disk init name = "auditdev",
    physname = "/dev/dsk/c2d0s4",
    size = "10M"
```

4　初始化要在其中放置安全性日志的设备：

```
disk init name = "auditlogdev",
    physname = "/dev/dsk/c2d0s5",
    size = "2M"
```

5　创建新的 sybsecurity 数据库：

```
create database sybsecurity on auditdev
    log on auditlogdev
```

6　将旧的 sybsecurity 数据库中的内容装载到新数据库中。将保留全局审计设置：

```
load database sybsecurity from "/remote/sec_file"
```

7　运行 online database，该命令将在必要时升级 sysaudits 和 sysauditoptions。

```
online database sybsecurity
```

8　使用所用平台的配置文档装载审计系统过程。

❖ **在 *sybsecurity* 中创建多个 *sysaudits* 表**

1　初始化要在其中放置其它表的设备：

```
disk init name = "auditdev2",
    physname = "/dev/dsk/c2d0s6",
    size = "10M"
```

2　将 sybsecurity 数据库扩展到步骤 1 中初始化的设备上：

```
alter database sybsecurity on auditdev2 = "2M"
```

3　运行 sp_addaudittable，在步骤 1 中初始化的设备上创建下一个 sysaudits 表：

```
sp_addaudittable auditdev2
```

4　对每个 sysaudits 表重复步骤 1 - 3。

设置审计追踪管理

若要有效地管理审计追踪：

1 确保审计安装后有两个或多个表，且每个表都在单独的设备上。如果不是，考虑添加审计表和设备。

2 编写一个阈值过程，并将其附加到每个审计表段。

3 为审计队列大小设置配置参数，并指明一旦当前审计表充满时应采取的相应措施。

以下各节假设您已安装的审计具有两个或更多个表，且每个表都位于单独的设备上。如果仅有一个设备用于审计表，可跳到第 265 页的 “单表审计”。

设置阈值过程

启用审计之前，建立一个阈值过程，以在当前表充满时自动切换审计表。

用于审计设备段的阈值过程应该：

- 使用 sp_configure 设置 current audit table 配置参数，使下一个空审计表成为当前表。
- 使用 insert...select 命令存档接近全满的审计表。

更改当前审计表

current audit table 配置参数建立 Adaptive Server 向其中写入审计行的表。作为系统安全员，您可以使用 sp_configure 按照以下语法更改当前审计表，其中， *n* 是一个整数，用以确定新的当前审计表：

```
sp_configure "current audit table", n
[, "with truncate"]
```

n 的有效值为：

- 1 表示 sysaudits_01， 2 表示 sysaudits_02，依此类推。
- 值 0 将告知 Adaptive Server 自动将下一个表设置为当前审计表。例如，如果系统中有三个审计表， sysaudits_01、 sysaudits_02 和 sysaudits_03，则 Adaptive Server 会将当前审计表设置为：
  - 2，如果当前审计表是 sysaudits_01
  - 3，如果当前审计表是 sysaudits_02
  - 1，如果当前审计表是 sysaudits_03

with truncate 选项指定如果新表不是空的， Adaptive Server 应截断该表。如果未指定此选项且新表不为空，则 sp_configure 将失败。

注释　如果 Adaptive Server 截断了当前的审计表，而且您没有将数据存档，则该表中的审计记录将丢失。因此，在使用 with truncate 选项之前，请将审计数据存档。

若要执行 sp_configure 以更改当前的审计表， sso_role 必须为活动状态。可以编写一个阈值过程来自动更改当前审计表。

存档审计表

可将 insert 与 select 一起使用，将审计数据复制到一个与 sybsecurity 中的审计表具有相同列的现有表中。

确保阈值过程可以成功将数据复制到另一数据库中的存档表中：

1　在不同于存放 sybsecurity 中审计表的另一设备上创建存档数据库。

2　创建一个存档表，表中的列应与 sybsecurity 审计表中的列相同。如果还没有这样的表，可使用 select into，通过在 where 子句中设置一个 false 条件来创建空表。例如：

```
use aud_db
go
select *
    into audit_data
    from sybsecurity.dbo.sysaudits_01
    where 1 = 2
```

where 条件始终为 false，因此将创建 sysaudits_01 的空副本。

存档数据库中的 select into/bulk copy 数据库选项必须设置为开启（使用 sp_dboption）后，才可以使用 select into。

阈值过程使用 sp_configure 改变审计表后，可以使用 insert 和 select 将数据复制到存档数据库的存档表中。此过程可以执行类似如下的命令：

```
insert aud_db.sso_user.audit_data
select * from sybsecurity.dbo.sysaudits_01
```

审计段的阈值过程示例

此样本阈值过程假定已为审计配置了三个表：

```
declare @audit_table_number int
/*
** Select the value of the current audit table
*/
select @audit_table_number = scc.value
from master.dbo.syscurconfigs scc, master.dbo.sysconfigures sc
where sc.config=scc.config and sc.name  = "current audit table"
/*
** Set the next audit table to be current.
** When the next audit table is specified as 0,
** the value is automatically set to the next one.
*/
exec sp_configure "current audit table", 0, "with truncate"
/*
** Copy the audit records from the audit table
** that became full into another table.
*/
if @audit_table_number = 1
    begin
        insert aud_db.sso_user.sysaudits
            select * from sysaudits_01
        truncate table sysaudits_01
    end
else if @audit_table_number = 2
    begin
        insert aud_db.sso_user.sysaudits
            select * from sysaudits_02
        truncate table sysaudits_02
    end
return(0)
```

将阈值过程附加到每个审计段

若要将阈值过程附加到每个审计表段，请使用 sp_addthreshold。

执行 sp_addthreshold 之前：

- 确定为安装配置的审计表数目及其设备段名称

- 具有对阈值过程中所有命令执行 sp_addthreshold 所需的权限和角色

警告！ sp_addthreshold 和 sp_modifythreshold 将进行检查，以确保只有直接被授予 sa_role 的用户才能添加或修改阈值。增加或修改阈值时，所有活动的系统定义角色都将作为有效角色插入到用户所登录的 systhresholds 表中。然而，触发阈值过程时，仅激活直接授予的角色。

审计表及其审计段

安装审计时， auditinit 显示每个审计表及其段的名称。 sysaudits_01 的段名为 “aud_seg1”， sysaudits_02 的段名为 “aud_seg2”，依此类推。如果使用 sybsecurity 作为当前数据库来执行 sp_helpsegment，则可以找到有关 sybsecurity 数据库中的段的信息。查明用户安装中的审计表数目的一种方法是执行以下 SQL 命令：

```
use sybsecurity
go
select count(*) from sysobjects
    where name like "sysaudit%"
go
```

通过执行以下 SQL 命令，获取有关审计表和 sybsecurity 数据库的额外信息：

```
sp_helpdb sybsecurity
go
use sybsecurity
go
sp_help sysaudits_01
go
sp_help sysaudits_02
go
  ...
```

所需角色和权限

若要执行 sp_addthreshold，您必须是数据库所有者或系统管理员。系统安全员应是 sybsecurity 数据库的所有者，因此应该能够执行 sp_addthreshold。除了能执行 sp_addthreshold，您还应该具有执行阈值过程中的所有命令的权限。例如，要为 current audit table 执行 sp_configure， sso_role 必须为活动状态。当阈值过程触发时， Adaptive Server 会尝试打开在执行 sp_addthreshold 时有效的所有角色和权限。

将阈值过程 audit_thresh 附加到三个设备段：

```
use sybsecurity
go
sp_addthreshold sybsecurity, aud_seg_01, 250, audit_thresh
sp_addthreshold sybsecurity, aud_seg_02, 250, audit_thresh
sp_addthreshold sybsecurity, aud_seg_03, 250, audit_thresh
go
```

当前审计表中保留的可用页少于 250 页时，样本阈值过程 audit_thresh 会接管控制。

有关添加阈值过程的详细信息，请参见 《系统管理指南：第二卷》中的第 16 章 “使用阈值管理可用空间”。

在样本阈值过程就位的情况下进行审计

启用审计后， Adaptive Server 会将所有审计数据写入最初的当前审计表 sysaudits_01。当 sysaudits_01 还剩不到 250 页就将写满时，阈值过程 audit_thresh 将触发。该过程将当前审计表切换到 sysaudits_02，而 Adaptive Server 会立即开始将新的审计记录写入 sysaudits_02。该过程还会将所有审计数据从 sysaudits_01 复制到 audit_db 数据库中的 audit_data 存档表。审计表的轮换以这种方式继续下去，不需要手工干预。

设置审计配置参数

为审计安装设置以下配置参数：

- audit queue size 设置内存中的审计队列中的记录数目。
- suspend audit when device full 确定在当前审计表完全写满时 Adaptive Server 的行为。仅当附加到当前表段的阈值过程运行不正常时，才会发生完全充满的情况。

设置审计队列的大小

缺省审计队列大小为 100 字节。审计队列池所消耗的内存量在 audit queue size 参数中定义，并包括数据缓冲区和内存池的开销。但是，池中的内存量可能会因版本和芯片体系结构而异。

使用 sp_configure 设置审计队列的长度。语法为：

sp_configure "audit queue size", [*value*]

value 是审计队列可容纳的记录数。最小值是 1，最大值是 65,535。例如，若要将审计队列大小设置为 300，请执行：

```
sp_configure "audit queue size", 300
```

有关设置审计队列大小和其它配置参数的详细信息，请参见 《系统管理指南：第一卷》中的第 5 章 “设置配置参数”。

设备已满时挂起审计

如果有两个或多个审计表，每个表都位于非主设备的单独设备上，而且每个审计表段都有阈值过程，则审计设备应绝不会充满。只有当阈值过程运行不正常时，才会发生 “全满”情况。使用 sp_configure 设置 suspend audit when device full 参数，以确定设备确实充满时所发生的情况。选择下列选项之一：

- 将导致审计事件的审计进程和所有用户进程挂起。系统安全员清除当前审计表后，恢复正常操作。
- 截断下一个审计表并开始使用它。这样，就可以不需要系统安全员的干预而继续进行正常操作。

使用 sp_configure to 设置此配置参数。sso_role 必须为活动状态。语法为：

```
sp_configure "suspend audit when device full",
    [0|1]
```

- 0 — 在当前审计表写满时，截断下一个审计表，并开始使用它作为当前审计表。如果将该参数设置为 0，审计进程将永不会挂起；但是，如果旧的审计记录尚未存档则会丢失。
- 1 （缺省值）— 挂起导致可审计事件的审计进程和所有用户进程。若要恢复正常操作，系统安全员必须登录并将一个空表设置为当前审计表。在此期间，系统安全员不能进行正常审计。如果系统安全员的操作在正常情况下会生成审计记录， Adaptive Server 会将一条错误消息和有关事件的信息发送到错误日志。

如果已将一个阈值过程附加到审计表段，则将 suspend audit when device full 设置为 1 （开启）。如果设置为 0 （关闭），则在阈值过程有机会存档审计记录之前， Adaptive Server 可能会截断已写满的审计表。

设置事务日志管理

本节描述有关管理 sybsecurity 中的事务日志的指导方针。

如果 trunc log on chkpt 数据库选项处于活动状态，则 Adaptive Server 每次执行自动 checkpoint 时，都将截断 syslogs。安装审计后，trunc log on chkpt 的值为 on，但可以使用 sp_dboption 来更改其值。

截断事务日志

如果为 sybsecurity 数据库启用了 trunc log on chkpt 选项，则不必担心事务日志会写满。 Adaptive Server 每次执行 checkpoint 时都会截断日志。如果启用此选项，则无法使用 dump transaction 来转储事务日志，但可使用 dump database 来转储数据库。

如果按照第 258 页的 “设置阈值过程”中的过程进行操作，审计表会自动存档到另一数据库的表中。可以对此存档数据库使用标准备份和恢复过程。

如果 sybsecurity 设备上出现故障，则可以重新装载数据库并继续进行审计。至多会丢失内存中的审计队列和当前审计表中的记录，因为存档数据库包含了所有其它审计数据。重新装载数据库后，使用 sp_configure with truncate 来设置和截断当前审计表。

如果转储数据库后没有改变过服务器范围的审计选项，则 sysauditoptions 中存储的所有审计选项在重新装载 sybsecurity 时都将自动恢复。如果没有恢复，可以运行脚本，以在重新开始审计前设置这些选项。

不截断的情况下管理的事务日志

如果使用 db_option 关闭 trunc log on chkpt，事务日志将会充满。您应计划将*最后机会阈值过程* 附加到事务日志段。当段中剩余的空间小于 Adaptive Server 自动计算的阈值量时，该过程将接管控制。该阈值量是备份事务日志所需的可用日志页数的一个估计值。

最后机会阈值过程的缺省名是 sp_thresholdaction，但只要 sa_role 为活动状态，就可使用 sp_modifythreshold 指定其它名称。

注释 sp_modifythreshold 将进行检查以确保激活了 “sa_role”。有关详细信息，请参见第 260 页的 “将阈值过程附加到每个审计段”。

Adaptive Server 未提供缺省过程，但《系统管理指南：第二卷》中的第 16 章“使用阈值管理可用空间”包含了最后机会阈值过程的示例。该过程应该执行 dump transaction 命令，后者将截断日志。事务日志达到最后机会阈值点时，正在运行的任何事务都将被挂起，直到有空间可用为止。由于 sybsecurity 数据库的 abort xact when log is full 选项总是设置为 false，因此会发生挂起。您不能更改此选项。

当 trunc log on chkpt 选项处于禁用状态时，可以对 sybsecurity 数据库使用标准备份和恢复过程，但要注意，恢复的数据库中的审计表与其在设备出现故障时的状态可能不同步。

启用和禁用审计

将 sp_configure 与 auditing 配置参数一起使用以启用或禁用审计。语法为：

```
sp_configure "auditing", [0 | 1 ]
```

- 1 — 启用审计。
- 0 — 禁用审计。

例如，若要启用审计，请输入：

```
sp_configure "auditing", 1
```

注释　当您启用或禁用审计时， Adaptive Server 会自动生成审计记录。请参见第 281 页的表 8-5 中的事件代码 73 和 74。

单表审计

Sybase 极力建议您不要对生产系统使用单个设备进行审计。如果仅使用一个审计表，则在存档审计数据和截断审计表时会消耗一定时间，在此期间将丢失写入的审计记录。仅使用一个审计表时没有方法能避免这种情况发生。

如果仅使用一个审计表，则该审计表很可能会充满。其后果取决于您对 suspend audit when device full 的设置情况。如果已将 suspend audit when device full 设置为打开，则审计进程将被挂起，导致审计事件的所有用户进程也被挂起。如果 suspend audit when device full 为关闭，审计表将被截断，将丢失审计表中的所有审计记录。

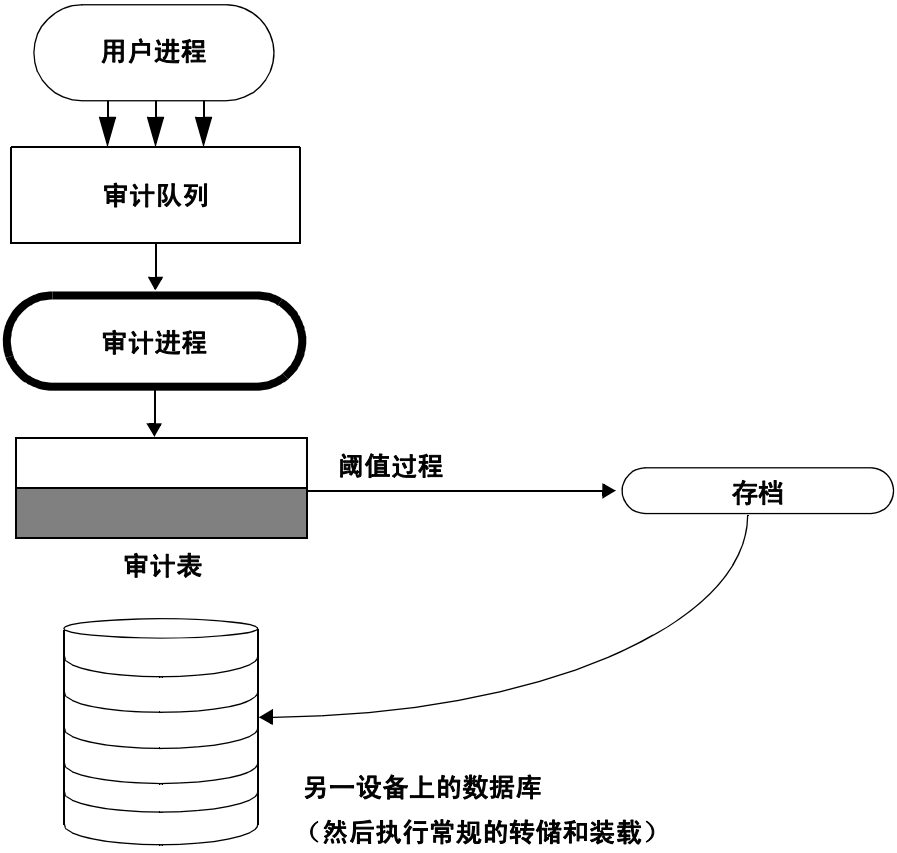
对于可接受少量审计记录丢失的非生产系统，如果不能腾出用于多个审计表的额外磁盘空间，或没有其它设备可用，您可以使用单个审计表审计。

使用单个审计表的过程与使用多个审计表的过程类似，不同之处如下所述：

- 安装时，仅指定一个系统表用于审计。
- 安装时，仅指定一个设备用于审计系统表。
- 为存档审计记录创建的阈值过程与使用多个审计表时创建的阈值过程不同。

图 8-2 说明了审计进程如何处理单个审计表。

图 8-2：使用单个审计表进行审计



建立和管理单表审计

单表审计与多表审计的配置步骤相同。

使用单表审计的审计进程

对于单表审计，其阈值过程应该：

- 用 insert 和 select 命令，将几乎充满的审计表存档到另一表中。
- 用 truncate table 命令，截断审计表以创建用于新审计记录的空间。

只有创建一个与审计表具有相同列的存档表之后，才可以存档审计记录。执行完这些步骤后，阈值过程就能使用 insert 和 select 将审计记录复制到存档表中。

下面是使用单个审计表的阈值过程示例：

```
create procedure audit_thresh as
/*
** copy the audit records from the audit table to
** the archive table
*/
insert aud_db.sso_user.audit_data
    select * from sysaudits_01
return(0)
go
/*
** truncate the audit table to make room for new
** audit records
*/
truncate table "sysaudits_01"
go
```

创建阈值过程后，将需要该过程附加到审计表段。有关说明，请参见第 260 页的 “将阈值过程附加到每个审计段”。

警告！在多处理器上，即使有审计表充满之前触发的阈值过程，审计表也可能会充满。例如，如果阈值过程运行于负载很重的 CPU 上，而执行可审计事件的用户进程运行于负载较小的 CPU 上，则触发阈值过程之前，审计表有可能充满。配置参数 suspend audit when device full 确定审计表充满时所发生的情况。有关设置该参数的信息，请参见第 263 页的 “设备已满时挂起审计”。

当前审计表充满时会发生什么？

当前审计表充满时：

1 审计进程试图将下一个审计记录插入该表中。此操作将失败，因此审计进程终止。将向错误日志中写入一条错误消息。

2 当用户尝试执行可审计事件时，由于审计无法继续，所以无法完成该事件。用户进程终止。没有试图执行可审计事件的用户不会受到影响。

3 如果启用了登录审计，则除了系统安全员外，其他人都不能登录到服务器。

4 如果在 sso_role 为活动状态时审计已执行的命令，则系统安全员将不能执行命令。

当前审计表充满时恢复

如果当前审计设备和审计队列都已满，则系统安全员不能执行审计。之后系统安全员执行的每个可审计事件都将向错误日志文件中发送一条警告消息。此消息指明日期和时间、说明审计已丢失的警告以及登录名、event 代码和通常存储在审计表的 extrainfo 列中的其它信息。

当前审计表已满时，系统安全员可以存档并截断审计表，如第 259 页的“存档审计表”中所述。系统管理员可以执行 shutdown 来停止服务器，然后重新启动服务器以重建审计。

如果审计系统异常终止，则系统安全员可在存档和截断当前审计表后关闭服务器。通常只有系统管理员才能执行 shutdown。

重新启动审计

如果审计进程因出现错误而强制终止，则可通过输入以下命令手动重新启动 sp_audit：

```
sp_audit restart
```

只要当前没有任何审计在运行，即可重新启动审计进程，但必须用 sp_configure “auditing” 1 启用审计进程。

设置全局审计选项

安装审计后，可使用 sp_audit 设置审计选项。 sp_audit 的语法为：

sp_audit *option*, *login_name*, *object_name* [,*setting*]

如果不带参数运行 sp_audit，它将提供选项的完整列表。有关 sp_audit 的详细信息，请参见 《参考手册：过程》。

注释　为服务器激活审计之前，不会进行审计。有关如何启动审计的信息，请参见第 265 页的 “启用和禁用审计”。

审计选项：类型和要求

能为 sp_audit 的 *login_name* 和 *object_name* 参数指定的值，取决于指定的审计选项类型：

- 全局选项应用于影响整个服务器的命令，例如启动服务器、磁盘操作命令和允许用户定义即席审计记录的命令。全局事件的选项设置存储在 sybsecurity..sysauditoptions 系统表中。
- 特定于数据库的选项应用于一个数据库。例如，更改数据库；将数据批量复制 (bcp in) 到数据库中；授予或撤消对数据库中对象的访问权；以及在数据库中创建对象，等等。特定于数据库的事件的选项设置存储在 master..sysdatabases 系统表中。
- 特定于对象的选项应用于一个对象。如选择、插入、更新或删除特定表或视图的行以及执行特定触发器或过程。特定于对象的事件的选项设置存储在相应数据库的 sysobjects 系统表中。
- 特定于用户的选项适用于特定的用户或系统角色。例如，特定用户对任何表或视图的访问，或当特定系统角色 （例如 sa_role）为活动状态时执行的所有操作。单个用户的选项设置存储在 master..syslogins 中。系统角色的设置存储在 master..sysauditoptions 中。
- 特定于角色的选项适用于特定的用户、组或系统角色，提供精细的与安全性相关的审计。“角色”审计选项对所有与角色相关的命令进行审计，审计选项 create、 alter 和 drop 用于审计角色定义命令，而 grant 和 revoke 用于审计向主体授予角色。为需要对象名参数的审计选项指定了 master 数据库。

表 8-2 显示：

- option 的有效值和每个选项的类型 （全局、特定于数据库、特定于对象或特定于用户）
- 每个选项的 *login_name* 和 *object_name* 参数的有效值
- 您设置审计选项时所在的数据库
- 您设置选项时被审计的命令或权限
- 每个选项的一个示例

所有选项的缺省值都是 “off”。

表 8-2：审计选项、要求和示例

选项 （选项类型）	*login_name*	*object_name*	设置选项时应在的数据库	将被审计的命令或权限
adhoc （特定于用户）	all	all	任何	允许用户使用 sp_addauditrecord
	此示例启用用户定义的即席审计记录： `sp_audit "adhoc", "all", "all", "on"`			
all （特定于用户）	一个登录名或角色	all	任何	特定用户或具有特殊角色的用户执行的所有操作
	此示例为 sa_role 处于活动状态的所有操作启用审计： `sp_audit "all", "sa_role", "all", "on"`			
alter （特定于数据库）	all	将被审计的数据库	任何	alter database、 alter role、 alter table
	此示例为 master 数据库中所有 alter database 和 alter table 执行启用审计： `sp_audit @option = "alter", @login_name = "all", @object_name = "master", @setting = "on"`			
bcp （特定于数据库）	all	将被审计的数据库	任何	bcp in
	此示例返回 pubs2 数据库中 bcp 审计的状态： `sp_audit "bcp", "all", "pubs2"` 如果没有为 *setting* 指定值， Adaptive Server 将返回指定选项的审计状态）			
bind （特定于数据库）	all	将被审计的数据库	任何	sp_bindefault、 sp_bindmsg、 sp_bindrule
	此示例为 planning 数据库禁用绑定审计： `sp_audit "bind", "all", "planning", "off"`			
cmdtext （特定于用户）	要审计的用户的登录名	all	任何	用户输入的 SQL 文本。 （不反映被审计的文本是否通过了权限检查。 *eventmod* 的值总为 1。）
	此示例为数据库所有者禁用文本审计： `sp_audit "cmdtext", "sa", "all", "off"`			

选项 （选项类型）	*login_name*	*object_name*	设置选项时应在的数据库	将被审计的命令或权限
create （特定于数据库）	all	将被审计的数据库	任何	create database、create table、create role、 create procedure、 create trigger、 create rule、 create default、 sp_addmessage、 create view、 create index、 create function
	注释　为 *object_name* 指定 master 以对 create database 进行审计。这样，还会对 master 中其它对象的创建进行审计。			
	此示例启用对 planning 数据库中成功对象创建操作的审计： `sp_audit "create", "all", "planning", "pass"` create database 的当前审计状态不会受影响，因为没有指定 master 数据库。）			
dbaccess （特定于数据库）	all	将被审计的数据库	任何	另一数据库对该数据库的任何访问
	此示例审计对 project 数据库的所有外部访问： `sp_audit "dbaccess", "all", "project", "on"`			
dbcc （全局）	all	all	任何	需要权限的所有 dbcc 命令
	此示例审计 dbcc 命令的所有执行： `sp_audit "dbcc", "all", "all", "on"`			
delete （特定于对象）	all	要审计的表或视图的名称，或者 default view 或 default table	该表或视图的数据库 （除 tempdb 之外）	从表中删除数据的 delete 命令、从视图中删除数据的 delete 命令
	此示例审计当前数据库中所有未来表的所有删除操作： `sp_audit "delete", "all", "default table", "on"`			
disk （全局）	all	all	任何	disk init、 disk refit、 disk reinit、 disk mirror、 disk unmirror、 disk remirror、 disk resize
	此示例审计服务器的所有磁盘操作： `sp_audit "disk", "all", "all", "on"`			
drop （特定于数据库）	all	将被审计的数据库	任何	drop database、 drop table、 drop role、 drop procedure、 drop index、 drop trigger、 drop rule、 drop default、 sp_dropmessage、 drop view、 drop function
	此示例审计 financial 数据库中所有未通过权限检查的删除命令： `sp_audit "drop", "all", "financial", "fail"`			

选项 （选项类型）	*login_name*	*object_name*	设置选项时应在的数据库	将被审计的命令或权限
dump （特定于数据库）	all	将被审计的数据库	任何	dump database、 dump transaction
	此示例审计 pubs2 数据库中的转储命令： `sp_audit "dump", "all", "pubs2", "on"`			
encryption_key （特定于数据库）	all	将被审计的数据库	任何	alter encryption key create encryption key drop encryption key sp_encryption
	此示例在 pubs2 数据库中审计所有上述命令： `sp_audit "encryption_key", "all", "pubs2", "on"`			
errors （全局）	all	all	任何	致命错误、非致命错误
	此示例审计整个服务器上的错误： `sp_audit "errors", "all", "all", "on"`			
errorlog	all	all	任何	sp_errorlog 或 errorlog_admin 函数
	此示例对尝试“更改日志”以转移到新 Adaptive Server 错误日志文件的操作进行审计： `sp_audit "errorlog", "all", "all", "on"`			
exec_procedure （特定于对象）	all	要审计的过程的名称或 default procedure	该过程的数据库（除 tempdb 之外）	execute
	此示例禁用对当前数据库中新过程的自动审计： `sp_audit "exec_procedure", "all", "default procedure", "off"`			
exec_trigger （特定于对象）	all	要审计的触发器的名称或 default trigger	该触发器的数据库 （除 tempdb 之外）	任何引发触发器的命令
	此示例对当前数据库中 trig_fix_plan 触发器的所有失败的执行进行审计： `sp_audit "exec_trigger", "all", "trig_fix_plan", "fail"`			
func_dbaccess （特定于数据库）	all	正在审计的数据库的名称	任何	使用以下函数对数据库进行的访问：curunreserved_pgs、 db_name、 db_id、 lct_admin、 setdbrepstat、 setrepstatus、 setrepdefmode、 is_repagent_enabled、 rep_agent_config、 rep_agent_admin
	此示例审计通过内置函数对 strategy 数据库的访问： `sp_audit @option="func_dbaccess", @login_name="all", @object_name = "strategy", @setting = "on"`			

选项 （选项类型）	*login_name*	*object_name*	设置选项时应在的数据库	将被审计的命令或权限
func_obj_access （特定于对象）	all	任何在 sysobjects 中具有一个条目的对象的名称	任何	使用以下函数对对象进行的访问：schema_inc、 col_length、 col_name、 data_pgs、 index_col、 object_id、 object_name、 reserved_pgs、 rowcnt、 used_pgs、 has_subquery
	此示例审计通过内置函数对 customer 表的访问： `sp_audit @option="func_obj_access", @login_name="all", @object_name = "customer", @setting = "on"`			
grant （特定于数据库）	all	要审计的数据库的名称	任何	grant
	此示例审计 planning 数据库中的所有授权： `sp_audit @option="grant", @login_name="all", @object_name = "planning", @setting = "on"`			
insert （特定于对象）	all	正在向其插入行的视图或表的名称，或者 default view 或 default table	该对象的数据库（除 tempdb 之外）	将数据插入到表中的 insert 命令、将数据插入到视图中的 insert 命令
	此示例对向当前数据库中的 dpt_101_view 视图插入数据的所有插入命令进行审计： `sp_audit "insert", "all", "dpt_101_view", "on"`			
install （特定于数据库）	all	将被审计的数据库	任何	install java
	此示例审计数据库 planning 中 Java 类的安装： `sp_audit "install", "all", "planning", "on"`			
load （特定于数据库）	all	将被审计的数据库	任何	load database、 load transaction
	此示例对 projects_db 数据库中所有已失败的、数据库和事务装载的执行进行审计： `sp_audit "load", "all", "projects_db", "fail"`			
login （全局）	all	all	任何	到 Adaptive Server 的任何登录
	此示例对所有已失败的、到服务器的登录尝试进行审计： `sp_audit "login", "all", "all", "fail"`			
login_locked （全局）	all	all	任何	
	此示例表明登录由于超过配置的失败登录尝试次数而被锁定： `sp_audit "login_locked", "all", "all", "on"`			
logout	all	all	任何	任何从 Adaptive Server 注销的操作
	此示例禁用对服务器注销操作的审计： `sp_audit "logout", "all", "all", "off"`			

选项 （选项类型）	*login_name*	*object_name*	设置选项时应在的数据库	将被审计的命令或权限
mount	all	all	任何	mount database
（全局）	此示例审计发出的所有 mount database 命令： `sp_audit "mount", "all", "all", "on"`			
password	all	all	任何	全局口令和登录策略选项的设置
	此示例对口令启用审计： `sp_audit "password", "all", "all", "on"`			
quiesce	all	all	任何	quiesce database
（全局）	此示例对 quiesce database 命令启用审计： `sp_audit "quiesce", "all", "all", "on"`			
reference	all	正在向其插入行的视图或表的名称，或者 default view 或 default table	任何	create table、 alter table
（特定于对象）	此示例禁用对 titles 表的引用创建的审计： `sp_audit "reference", "all", "titles", "off"`			
remove	all	all	任何	审计 Java 类的删除
（特定于数据库）	此示例审计数据库 planning 中 Java 类的删除： `sp_audit "remove", "all", "planning", "on"`			
revoke	all	将被审计的数据库	任何	revoke
（特定于数据库）	此示例禁用对 payments_db 数据库中 revoke 执行的审计： `sp_audit "revoke", "all", "payments_db", "off"`			
rpc	all	all	任何	远程过程调用 （内部或外部）
（全局）	此示例审计对服务器外部或内部进行的所有远程过程调用： `sp_audit "rpc", "all", "all", "on"`			
security	all	all	任何	全服务器范围内与安全性相关的事件。请参见表 8-5 中的 “安全性” 选项。
（全局）	此示例对服务器中全服务器范围内的与安全性相关的事件进行审计： `sp_audit "security", "all", "all", "on"`			
select	all	正在向其插入行的视图或表的名称，或者 default view 或 default table	该对象的数据库（除 tempdb 之外）	从表中选择数据的 select 命令、从视图中选择数据的 select 命令
（特定于对象）	此示例对从当前数据库中的 customer 表选择数据但失败的所有选择命令进行审计： `sp_audit "select", "all", "customer", "fail"`			

选项 （选项类型）	*login_name*	*object_name*	设置选项时应在的数据库	将被审计的命令或权限
setuser	all	all	任何	setuser
（特定于数据库）	此示例审计 projdb 数据库中 setuser 的所有执行： `sp_audit "setuser", "all", "projdb", "on"`			
table_access	要审计的用户的登录名。	all	任何	表中的 select、 delete、 update 或 insert 访问
（特定于用户）	此示例审计由登录名 “smithson”执行的所有表访问： `sp_audit "table_access", "smithson", "all", "on"`			
transfer_table	all	all	任何	全服务器范围内的选项。不显示在 sysauditoptions 中。
（全局）	此示例对服务器中全服务器范围内的与传输相关的事件进行审计： `sp_audit "transfer_table", "tdb1.table1", "all", "on"`			
truncate	all	将被审计的数据库	任何	truncate table
（特定于数据库）	此示例审计 customer 数据库中的所有表截断： `sp_audit "truncate", "all", "customer", "on"`			
unbind	all	将被审计的数据库	任何	sp_unbindefault、 sp_unbindrule、 sp_unbindmsg
（特定于数据库）	此示例对 master 数据库中所有失败的解除绑定尝试进行审计： `sp_audit "unbind", "all", "master", "fail"`			
unmount	all	all	任何	unmount database
（全局）	此示例对使用任何数据库卸载或创建清单文件的所有尝试进行审计： `sp_audit "unmount", "all", "all", "on"`			
update	all	指定要审计的对象的名称， default table 或 default view	该对象的数据库（除 tempdb 之外）	对表执行的 update 命令、对视图执行的 update 命令
（特定于对象）	此示例对用户更新当前数据库中 projects 表的所有尝试进行审计： `sp_audit "update", "all", "projects", "on"`			
view_access	要审计的用户的登录名	all	任何	对视图执行的 select、 delete、 insert 或 update
（特定于用户）	此示例禁用用户 “joe”的视图审计： `sp_audit "view_access", "joe", "all", "off"`			

设置审计选项的示例

假定用户要对 company_operations 数据库中 projects 表和该数据库中所有新表上所有失败了的删除进行审计。对 projects 表使用特定于对象的 delete 选项，对数据库中的所有未来表使用 default table。在执行 sp_audit 以设置特定于对象的审计选项之前，您必须处于对象的数据库中：

```
sp_audit "security", "all", "all", "fail"
```

对于此示例，应执行：

```
use company_operations
go
sp_audit "delete", "all", "projects", "fail"
go
sp_audit "delete", "all", "default table",
"fail"
go
```

角色定义审计

示例 1. 为角色变更启用审计：

```
sp_audit "alter", "all", "master", "pass"
```

示例 2. 为成功的角色创建启用审计：

```
sp_audit "alter", "all", "master", "on"
```

示例 3. 此示例禁用删除角色的审计：

```
sp_audit "drop", "all", "master", "off"
```

示例 4. 禁用授予角色的审计：

```
sp_audit "grant", "all", "master", "off"
```

使用 grant 或 role 审计选项执行审计，生成 AUD_EVT_UDR_CMD (85) 事件审计记录。

示例 5. 启用撤消规则的审计：

```
sp_audit "revoke", "all", "master", "on"
```

使用 revoke 或 role 审计选项执行审计，生成 AUD_EVT_UDR_CMD (85) 事件审计记录。

隐藏系统存储过程和命令口令参数

如果配置并启用了审计，且设置了 sp_audit 选项 'cmdtext'，则审计日志中包含的审计记录里的系统存储过程和命令口令参数就会被替换为固定长度的星号串。

例如，当启用审计并设置 sp_audit cmdtext 后，执行以下命令：

```
alter login johnd with password oldpasswd modify
password 'newpasswd'
```

此命令将导致类似如下的输出：

```
alter login johnd with password ****** modify password
'******'
```

这样做可以保护口令不会被可以访问审计日志的其他人看到。

确定当前审计设置

若要确定给定选项的当前审计设置，请使用 sp_displayaudit。语法为：

```
sp_displayaudit [procedure | object | login | database | global |
    default_object | default_procedure [, name]]
```

有关详细信息，请参见 《参考手册：过程》中的 sp_displayaudit。

向审计追踪中添加用户指定的记录

sp_addauditrecord 允许用户向审计追踪中输入注释。语法为：

```
sp_addauditrecord [text] [, db_name] [, obj_name]
    [, owner_name] [, dbid] [, objid]
```

所有参数都是可选的。

- *text* — 是要增加到 extrainfo 审计表中的消息文本。
- *db_name* — 是记录中引用的数据库的名称，它会被插入到当前审计表的 dbname 列中。
- *obj_name* — 是记录中引用的对象的名称，它会被插入到当前审计表的 objname 列中。
- *owner_name* 是记录中引用的对象的所有者，它会被插入到当前审计表的 objowner 列中。

- *dbid* — 是一个整数值，表示 db_name 的数据库 ID 号，此数值会被插入到当前审计表的 dbid 列中。不要将其置于引号中。
- *objid* — 是一个整数值，表示 obj_name 的对象 ID 号。不要将该值用引号引上。*objid* 会被插入到当前审计表的 objid 列中。

在下列情况下，可使用 sp_addauditrecord：

- 对 sp_addauditrecord 具有执行权限。
- 使用 sp_configure 激活了审计配置参数。
- adhoc 审计选项通过 sp_audit 启用。

缺省情况下，只有系统安全员和 sybsecurity 的数据库所有者才能使用 sp_addauditrecord。可以将执行该命令的权限授予其他用户。

增加用户定义的审计记录的示例

下面的示例向当前审计表中增加一条记录。文本部分将输入到当前审计表的 extrainfo 列中；“corporate”将输入到 dbname 列中；“payroll”将输入到 objname 列中；“dbo”将输入到 objowner 列中；“10”将输入到 dbid 列中；“1004738270”将输入到 objid 列中：

```
sp_addauditrecord "I gave A. Smith permission to view
the payroll table in the corporate database. This
permission was in effect from 3:10 to 3:30 pm on
9/22/92.", "corporate", "payroll", "dbo", 10,
1004738270
```

下面的示例仅将信息插入当前审计表的 extrainfo 和 dbname 列中：

```
sp_addauditrecord @text="I am disabling auditing
briefly while we reconfigure the system",
@db_name="corporate"
```

查询审计追踪

若要查询审计追踪，请使用 SQL 来选择和总结审计数据。如果按照第 258 页的 “设置审计追踪管理”中描述的过程进行操作，审计数据将被自动存档到另一数据库中的一个或多个表中。例如，假定审计数据位于 audit_db 数据库中名为 audit_data 的表中。若要选择由 “bob”于 1993 年 7 月 5 日所执行任务的审计记录，请执行：

```
use audit_db
go
select * from audit_data
   where loginname = "bob"
   and eventtime like "Jul 5% 93"
go
```

以下命令请求具有系统安全员活动角色的用户在 pubs2 数据库中所执行的命令的审计记录：

```
select * from audit_data
   where extrainfo like "%sso_role%"
   and dbname = "pubs2"
go
```

下面的命令请求所有表截断 （事件 64）的审计记录：

```
select * from audit_data
   where event = 64
go
```

要使用审计事件的名称来查询审计追踪，请使用 audit_event_name 函数。例如，若要请求所有数据库创建事件的审计记录，请输入：

```
select * from audit_data where audit_event_name(event)
   = "Create Database"
go
```

了解审计表

只有系统安全员才能访问系统审计表，该用户可通过执行 SQL 命令来读取这些表。只允许对系统审计表执行 select 和 truncate 命令。

表 8-3 介绍了所有审计表中的列。

表 8-3：每个审计表中的列

列名	数据类型	说明
event	smallint	审计的事件的类型。请参见第 281 页的表 8-5。
eventmod	smallint	有关被审计的事件的详细信息。指示被审计的事件是否通过了权限检查。可能的值有： • 0 = 此事件无修饰符。 • 1 = 事件通过了权限检查。 • 2 = 事件未通过权限检查。
spid	smallint	引起写入审计记录的进程的 ID。
eventtime	datetime	审计事件发生的日期和时间。
sequence	smallint	单个事件中记录的序列号。某些事件需要多个审计记录。
suid	smallint	执行审计事件的用户的服务器登录 ID。
dbid	int null	被审计的事件发生时所在的数据库 ID，或者对象、存储过程或触发器（取决于事件的类型）所在的数据库 ID。
objid	int null	被访问的对象、存储过程或触发器的 ID。
xactid	binary(6) null	包含审计事件的事务的 ID。对于多数据库事务，这是该事务在其中启动的数据库中的事务 ID。
loginname	varchar(30) null	与 suid 相对应的登录名。
dbname	varchar(30) null	与 dbid 相对应的数据库名。
objname	varchar(30) null	与 objid 相对应的对象名。
objowner	varchar(30) null	objid 的所有者名。
extrainfo	varchar(255) null	有关审计事件的其它信息。该列包含一系列用分号分开的项。有关详细信息，请参见第 281 页的 “读取 extrainfo 列”。
nodeid	tinyint	发生事件的集群中的服务器 nodeid。

读取 *extrainfo* 列

extrainfo 列包含一系列用分号分隔的数据。数据按以下种类组织。

表 8-4：extrainfo 列中的信息

位置	类别	说明
1	角色	由空格分开的活动角色列表。
2	关键字或选项	用于事件的关键字或选项的名称。例如，对于 alter table 命令，可能已经使用了 add column 或 drop constraint 选项。如果列出了多个关键字或选项，则用逗号分开。
3	以前值	如果事件导致了值的更新，则此项包含更新之前的值。
4	当前值	如果事件导致了值的更新，则此项包含新值。
5	其它信息	为事件记录的其它与安全性相关的信息。
6	代理信息	初始登录名 （如果在 set proxy 有效时发生该事件）。
7	主管名	来自基础安全性机制的主管名 （如果用户的登录名是安全缺省登录名，并且用户通过统一登录名登录到 Adaptive Server）。如果没有使用安全缺省登录名，则此项的值为 NULL。

下面的示例显示了改变审计配置参数事件的 extrainfo 列条目。

```
sso_role;suspend audit when device full;1;0;;ralph;
```

此条目表示系统安全员已将 suspend audit when device full 从 1 更改为 0。此条目没有 “其它信息”。第 6 个类别指出用户 “ralph”正在使用代理登录。没有提供主管名。

此审计记录中的其它字段提供了其它有关信息。例如，该记录包含服务器用户 ID (suid) 和登录名 (loginname)。

表 8-5 列出了 event 列中显示的值 （按 sp_audit 选项排列）。“extrainfo 中的信息”列按照表 8-4 中的类别介绍了可能显示在审计表的 extrainfo 列中的信息。

表 8-5：事件和 extrainfo 列中的值

审计选项	将被审计的命令或权限	event	extrainfo 中的信息
（不受选项控制的自动被审计的事件）	使用以下命令启用审计：sp_configure auditing	73	—
（不受选项控制的自动被审计的事件）	使用以下命令禁用审计：sp_configure auditing	74	—
解锁管理员的帐户	使用以下命令禁用审计：sp_configure auditing	74	—
adhoc	用户定义的审计记录	1	用 sp_addauditrecord 的 text 参数填充 extrainfo

审计选项	将被审计的命令或权限	event	extrainfo 中的信息
alter	alter database	2	*子命令关键字：* alter maxhold alter size inmemory
	alter...modify owner *name_in_db*	124	*子命令关键字：* • 对于用户定义的类型：*owner. obj_name name_in_db* preserve permissions （如果指定该选项）。 • 对于对象：*name_in_db* preserve permission （如果指定该选项）。
	alter...modify owner *login_name* as concrete_owner	124	子命令关键字： 不适用于用户定义的数据类型：对于对象： *login_name* preserve permissions （如果指定该选项）。
	alter table	3	*子命令关键字：* add/drop/modify column replace columns replace decrypt default replace/add decrypt default add constraint drop constraint 如果添加了一个或多个加密列，则 extrainfo 将包含以下内容，其中 *keyname* 是键的完全限定名： add/drop/modify column *column1*/*keyname1*, [,*column2*/*keyname2*]
bcp	bcp in	4	—
bind	sp_bindefault	6	*其它信息：*缺省值的名称
	sp_bindmsg	7	*其它信息：*消息 ID
	sp_bindrule	8	*其它信息：*规则的名称
all、 create	create database	9	关键字或选项：inmemory
cmdtext	所有命令	92	命令的全文本，由客户端发送

审计选项	将被审计的命令或权限	event	extrainfo 中的信息
create	create database	9	—
	create default	14	—
	create procedure	11	—
	create rule	13	—
	create table	10	对于加密列， extrainfo 包含列名和密钥名。EK *column1*/*keyname1*[,*column2 keyname2*] 其中， EK 为前缀，用于指示后续信息将引用加密密钥；而 *keyname* 则是密钥的完全限定名。
	create trigger	12	—
	create view	16	—
	create index	104	*其它信息：*索引的名称
	create function	97	—
	sp_addmessage	15	*其它信息：*消息号
dbaccess	任何用户对数据库进行的任何访问	17	*关键字或选项：* use cmd outside reference
dbcc	dbcc （所有关键字）	81	*关键字或选项：*任意 dbcc 关键字 （例如 checkstorage）及该关键字的选项。
delete	从表中删除数据的 delete 命令	18	*关键字或选项：*delete
	从视图中删除数据的 delete 命令	19	*关键字或选项：*delete
disk	disk init	20	*关键字或选项：*disk init *其它信息：*磁盘的名称
	disk mirror	23	*关键字或选项：*disk mirror *其它信息：*磁盘的名称
	disk refit	21	*关键字或选项：*disk refit *其它信息：*磁盘的名称
	disk reinit	22	*关键字或选项：*disk reinit *其它信息：*磁盘的名称
	disk release	87	*关键字或选项：*disk release *其它信息：*磁盘的名称
	disk remirror	25	*关键字或选项：*disk remirror *其它信息：*磁盘的名称
	disk unmirror	24	*关键字或选项：*disk unmirror *其它信息：*磁盘的名称
	disk resize	100	*关键字或选项：*disk resize *其它信息：*磁盘的名称

审计选项	将被审计的命令或权限	event	extrainfo 中的信息
drop	drop database	26	—
	drop default	31	—
	drop procedure	28	—
	drop table	27	—
	drop trigger	29	—
	drop rule	30	—
	drop view	33	—
	drop index	105	*其它信息*：索引名
	drop function	98	—
	sp_dropmessage	32	*其它信息*：消息号
dump	dump database	34	—
	dump transaction	35	—

encryption_key	sp_encryption	106	如果第一次设置口令：
			`ENCR_ADMIN system_encr_passwd password ********`
			如果后来更改口令：
			`ENCR_ADMIN system_encr_passwd password ******** ********`

	create encryption key	107	关键字包含： algorithm name-bitlength/IV [random|NULL]/pad [random |NULL] user/system 例如： `AES-128/IV RANDOM/PAD NULL USER default/not default`
	alter encryption key	108	
	drop encryption key	109	
	AEK modify encryption	118	modify encryption with user passwd | for user *username* {with login passwd | with user passwd | with *keyvalue*} [for recovery 请注意，*keyvalue* 只会为 alter encryption key modify encryption 的重复项显示。例如，当用户 “stephen”修改其密钥副本时，将保存以下信息： `MODIFY ENCRYPTION for user stephen WITH USER PASSWD`

审计选项	将被审计的命令或权限	event	extrainfo 中的信息
	AEK add encryption	119	add encryption for user *user_name* for login association | recovery|with keyvalue] 请注意，*keyvalue* 只会为 alter encryption key add encryption 的重复项显示。
	alter encryption key drop encryption	120	drop encryption [for recovery | for user *user_name* 请参见 《加密列用户指南》。
	alter encryption key modify owner	121	modify owner [new owner *user_name*] 请参见 《加密列用户指南》。
	alter encryption key recover key	122	recovery key [with *key_value*] with *keyvalue* 只会在 alter encryption key 的复制过程中使用 请参见 《加密列用户指南》。
errorlog	errorlog 或 errorlog_admin 函数	127	将记录传递到 errorlog_admin 的参数以标识子命令：errorlog_admin (param1, param2,...)。
errors	致命错误	36	*其它信息：错误号 . 严重性 . 状态*
	非致命错误	37	*其它信息：Error number.Severity.State*
exec_procedure	执行过程	38	*其它信息：*所有输入参数
exec_trigger	执行触发器	39	—
func_obj_access、func_dbaccess	通过 Transact-SQL 函数对对象和数据库的访问。（必须为 sa_role 启用审计才能审计函数）。	86	—
grant	grant	40	包含完整的命令文本 （如果可用）。否则，包含被授予者和命令类型。
insert	向表中插入数据的 insert 命令	41	*关键字或选项：* • 如果使用 insert：insert • 如果使用 select into：insert into，后跟完全限定对象名
	向视图中插入数据的 insert 命令	42	*关键字或选项*：insert
install	install	93	—
load	load database	43	—
	load transaction	44	—
login	到服务器的任何登录	45	*其它信息*： • 从中执行登录的计算机的主机名和 IP 地址。 • *错误号 . 严重性 . 状态*（对于失败的登录）。

审计选项	将被审计的命令或权限	event	extrainfo 中的信息
login_locked	由于超过配置的失败登录尝试次数而被锁定的登录	112	
logout	服务器的任何注销	46	*其它信息*：主机名
mount	mount database	101	—
password	sp_passwordpolicy 及其所有操作（list 除外）	115	sp_passwordpolicy 的参数
quiesce	quiesce database	96	—
reference	表的引用的创建	91	*关键字或选项*：reference *其它信息*：引用表的名称
remove	remove java	94	—
revoke	revoke	47	包含完整的命令文本 （如果可用）。否则，包含被授予者和命令类型。
rpc	来自另一台服务器的远程过程调用	48	*关键字或选项*：客户端程序的名称 *其它信息*：服务器名，即从其执行 RPC 的计算机的主机名。
	到另一台服务器的远程过程调用	49	*关键字或选项*：过程名称
role locked	角色设置 / 取消设置	133	角色名和锁定原因： • 角色已由 suid 通过执行 alter role *rolename* lock 手动锁定 • 角色已由 Adaptive Server 锁定，因为角色激活尝试的失败次数达到了 max failed_logins 限制
security	connect to （仅限 CIS）	90	*关键字或选项*：connect to
	online database	83	—
	proc_role 函数 （从系统过程中执行）	80	*其它信息*：所需的角色
	由 sso 重新生成口令	76	*关键字或选项*：设置 SSO 口令 *其它信息*：登录名
	角色切换	55	*以前值*：on 或 off *当前值*：on 或 off *其它信息*：正在设置的角色的名称
	服务器启动	50	*其它信息*： -d*masterdevicename* -i*interfaces file path* -S*servername* -e*errorfilename*
	sp_webservices	111	*关键字或选项*：deploy （如果配置一项 Web 服务）。 deploy_all （如果配置所有 Web 服务）

审计选项	将被审计的命令或权限	event	extrainfo 中的信息
	sp_webservices	111	*关键字或选项：*undeploy （如果取消配置一项 Web 服务）。 undeploy_all （如果取消配置所有 Web 服务）
	服务器关闭	51	*关键字或选项：*shutdown
	set proxy 或 set session authorization	88	*以前值：*以前的 suid *当前值：*新的 suid
	sp_configure	82	*关键字或选项：*SETCONFIG *其它信息：* • 如果正在设置某一参数：配置参数的数量 • 如果正在使用配置文件设置参数：配置文件的名称
	sp_ssladmin （启用管理）	99	关键字包含 SSL_ADMIN addcert（如果添加认证）。
	审计表访问	61	—
	create login、 drop login	103	*关键字或选项：*create login、 drop login
	create、 drop、 alter、 grant 或 revoke role	85	*关键字或选项：*create、 drop、 alter、 grant 或 revoke role
	内置函数	86	*关键字或选项：*函数的名称
	要审计的安全命令或访问，尤其是使用 -u 选项启动 Adaptive Server 来解锁管理员帐户。	95	其它信息包含 “解锁管理员帐户”
	对 LDAP state changes 的更改	123	*关键字或选项：*主 URL 状态和辅助 URL 状态 • 以前值 • 当前值 其它信息指示状态更改是自动发生还是由于手动输入的命令而发生的。
	系统或 sp_passwordpolicy 为网络口令加密进行的非对称密钥对重新生成	117	extrainfo 中的信息
select	从表中选择数据的 select 命令	62	*关键字或选项：* select into select readtext
	从视图中选择数据的 select 命令	63	*关键字或选项：* select into select readtext

审计选项	将被审计的命令或权限	event	extrainfo 中的信息
setuser	setuser	84	*其它信息*：正在设置的用户的名称
table_access	delete	18	*关键字或选项*：delete
	insert	41	*关键字或选项*：insert
	select	62	*关键字或选项*： select into select readtext
	update	70	*关键字或选项*： update writetext
truncate	truncate table	64	—
transfer_table	transfer table	136	transfer table
unbind	sp_unbindefault	67	—
	sp_unbindmsg	69	—
	sp_unbindrule	68	—
unmount	unmount database	102	—
	create manifest file	116	extrainfo 中的信息
update	对表执行的 update 命令	70	*关键字或选项*： update writetext
	对视图执行的 update 命令	71	*关键字或选项*： update writetext
view_access	delete	19	*关键字或选项*：delete
	insert	42	*关键字或选项*：insert
	select	63	*关键字或选项*： select into select readtext
	update	71	*关键字或选项*： update writetext

表 8-6 列出了 event 列中显示的值 （按审计事件排列）。

表 8-6：审计事件值

审计事件 ID	命令名	审计事件 ID	命令名
1	ad hoc audit record	62	select table
2	alter database	63	select view
3	alter table	64	truncate table
4	bcp in	65	保留
5	保留	66	保留
6	bind default	67	unbind default
7	bind message	68	unbind rule
8	bind rule	69	unbind message
9	create database	70	update table
10	create table	71	update view
11	create procedure	72	保留
12	create trigger	73	启用审计
13	create rule	74	禁用审计
14	create default	75	保留
15	create message	76	SSO 更改口令
16	create view	77	保留
17	access to database	78	保留
18	delete table	79	保留
19	delete view	80	执行的角色检查
20	disk init	81	dbcc
21	disk refit	82	配置
22	disk reinit	83	online database
23	disk mirror	84	setuser 命令
24	disk unmirror	85	UDR 命令
25	disk remirror	86	内置函数
26	drop database	87	磁盘释放
27	drop table	88	set SSA 命令
28	drop procedure	89	kill 或 terminate 命令
29	drop trigger	90	connect
30	drop rule	91	引用
31	drop default	92	命令文本
32	drop message	93	JCS install 命令
33	drop view	94	JCS remove 命令
34	dump database	95	解锁管理员帐户

审计事件 ID	命令名	审计事件 ID	命令名
35	dump transaction	96	quiesce database
36	致命错误	97	create SQLJ 函数
37	非致命错误	98	drop SQLJ 函数
38	存储过程的执行	99	SSL 管理
39	触发器的执行	100	disk resize
40	grant	101	mount database
41	insert table	102	unmount database
42	insert view	103	login 命令
43	load database	104	create index
44	load transaction	105	drop index
45	login	106	sp_encryption （加密列管理）
46	logout	107	create encryption key
47	revoke	108	Alter Encryption Key as/not default
48	rpc in	109	drop encryption key
49	rpc out	110 111	deploy user-defined web services undeploy user defined web services
50	server boot	112	登录已锁定
51	服务器关闭	113	quiesce hold security
52	保留	114	quiesce release
53	保留	115	口令管理
54	保留	116	create manifest file
55	角色切换	117	regenerate keypair
56	保留	118	alter encryptin key modify encryption
57	保留	119	alter encryption key add encryption
58	保留	120	alter encryption key drop encryption
59	保留	121	alter encryption key modify owner
60	保留	122	alter encryption key for key recovery
61	对审计表的访问	123	LDAP 状态更改
		124	alter...modify owner
		127	错误日志管理
		136	transfer table

监控失败登录尝试次数

当登录帐户由于失败登录尝试次数超出配置次数而被锁定时，将记录审计选项 login_locked 和事件 Locked Login (value 112)。设置了审计选项 login_locked 即会启用此事件。要设置 login_locked，请输入：

```
sp_audit "login_locked","all","all","ON"
```

如果审计表已满，并且无法记录事件，则会向错误日志发送包含相关信息的消息。

审计记录中包括主机名和网络 IP 地址。监控 Locked Login 事件（编号 112）的审计日志有助于确定对登录帐户的攻击。

审计登录失败

尽管客户端应用程序可能由于多种原因而登录失败，但 Adaptive Server 不会向这些应用程序提供任何有关登录失败的详细信息。这样可以避免将相关信息提供给企图破译口令或以其它方式违反 Adaptive Server 的鉴定机制的恶意用户。

但是，作为系统管理员，详细信息有助于诊断 Adaptive Server 的管理或配置问题，并可帮助安全员调查是否有人试图违反安全性机制。

以下命令可启用对所有登录失败的审计：

```
sp_audit "login", "all", "all", "fail"
```

为防止对信息的不适当的使用，只有被授予了 SSO 角色的用户才能访问包含此敏感信息的审计追踪信息。

Adaptive Server 将审计以下情况下的登录失败：

- 对于作为 Windows 服务启动的 Adaptive Server，如果 Sybase SQLServer 服务已暂停（例如，由 Microsoft Management Console for Services 暂停）。
- 如果远程服务器尝试建立服务器到服务器 RPC 的节点处理器，但资源不足 （或此处列出的任何其它情况）导致节点处理器初始化失败。
- 对 Adaptive Server for Windows 使用 “受托登录”或 “统一登录”配置，但指定的用户不是受托管理员 （即鉴定失败）。
- Adaptive Server 不支持客户端请求的 SQL 接口。
- 用户试图在单用户模式下登录到 Adaptive Server。在单用户模式下，只允许一个具有 sa_role 的用户登录到 Adaptive Server。其它登录会被阻止，即使用户具有 sa_role。

- 主数据库中的 syslogins 表无法打开，表明主数据库具有内部错误。
- 客户端试图远程登录，但 sysremotelogins 无法打开，或指定用户帐户没有任何条目且不存在任何 Guest 帐户。
- 客户端试图远程登录，虽然它在 sysremotelogins 中找到引用指定用户的本地帐户的条目，但引用的本地帐户不存在。
- 客户端程序请求安全会话 （例如 Kerberos 鉴定），但由于以下原因无法建立安全会话：
  - Adaptive Server 安全子系统未在启动时初始化。
  - 没有足够的内存资源来存储分配的结构。
  - 鉴定协商失败。
- 未找到针对指定用户的鉴定机制。
- 指定的口令不正确。
- syslogins 不包含指定登录所需的条目。
- 登录帐户被锁定。
- Adaptive Server 已达到了它对用户连接数量的限制。
- 设置了配置参数 unified login required，但相应的安全子系统未鉴定登录。
- Adaptive Server 的网络缓冲区不可用，或请求的包大小无效。
- 客户端应用程序请求基于主机的通信套接字连接，但没有用于基于主机的通信缓冲区的内存资源。
- 正在关闭，但指定用户没有 sa 角色。
- Adaptive Server 无法打开缺省数据库进行登录，且此登录没有对 master 数据库的访问权限。
- 客户端发出了高可用性登录故障切换请求，但高可用性子系统没有对此登录的高可用性会话，或此登录无法等待故障切换完成。
- 客户端请求高可用性登录设置，但高可用性子系统无法创建会话，或无法完成对高可用性会话的 TDS 协议协商。
- Adaptive Server 无法设置 tempdb 进行登录。
- 检测到 TDS 登录协议错误。

索引

符号

英文

A

B

C

D

F

G

H

J

K

L

M

N

P

Q

R

S

T

W

X

Y

Z