SYBASE®
An SAP® Company

セキュリティ管理ガイド

Adaptive Server® Enterprise

15.7

ドキュメント ID：DC01813-01-1570-01
改訂：2011 年 9 月

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

目次

第 1 章

セキュリティの概要

セキュリティの概要

情報は、おそらく企業にとって最大の資産です。他のすべての資産と同じように、情報も保護する必要があります。システム管理者は、データベースに格納されている情報を保護するための最善の方法と、情報にアクセスできる人物を決定する必要があります。個々のデータベース・サーバは、強力ではあるが柔軟性のあるセキュリティのサポートを必要とします。

ユーザとユーザがアクセスするデータは、世界中に分散し、それらを結ぶネットワークは必ずしも常に信頼できるものではありません。このような環境では、機密データとトランザクションの機密性と整合性を保持することは重要な意味を持ちます。

情報は、情報を必要とする人物が情報を必要とするときに入手できる場合にのみ役に立ちます。ダイナミックに変化する複雑なビジネス関係の中では、権限のあるユーザだけが情報を入手できることは極めて重要です。

情報セキュリティの概要

次に、企業のセキュリティを考慮する際の一般的なガイドラインを示します。

- 重要な情報の機密性が保たれていること ― 誰がどの情報にアクセスできるかを決めます。
- システムの整合性が保たれていること ― サーバは、ルールと制約を使用して、情報の正確性と完全性が保たれることを保証する必要があります。

- 必要な情報が入手可能であること – すべての保護手段が機能している場合でも、情報にアクセスする人物が必要に応じて情報を入手できる必要があります。

組織が何を保護したいのか、そして外の世界が組織から何を求めているかを特定します。

- 情報資産と、それらが危険にさらされたり問題が発生したりした場合のセキュリティ上のリスクを把握する。
- 組織と情報資産に適用される法律、法令、規制、契約上の取り決めをすべて確認し、理解する。
- 組織のビジネス・プロセスと、情報資産に課せられている要件を特定し、セキュリティ上のリスクとそれらの間で運用上のバランスをとる。

セキュリティの要件は、将来にわたって変わる可能性があります。セキュリティ要件が常に組織の要求を反映しているかどうかを確認するために、セキュリティ要件の評価を定期的に繰り返し実施してください。

情報セキュリティに関する決定事項が明記されている情報セキュリティ・ポリシー・ドキュメントを作成した後に、組織のセキュリティ目標と合致する一連の管理手段と方針を設定します。

Adaptive Server® には、企業のセキュリティ・ポリシーを強制するのに役立つ一連のセキュリティ機能が含まれています。Adaptive Server のセキュリティ機能の詳細については、「第 2 章 Adaptive Server のセキュリティ管理について」を参照してください。

情報セキュリティ規格

Adaptive Server は、CCEVS (Common Criteria Evaluation and Validation Scheme) の規定に従って評価されました。また、Adaptive Server では、暗号化機能を実装するため、FIPS 140-2 認定モジュールを使用しています。

この項では、これらの認定について説明します。

Common Criteria 設定評価

Common Criteria for Information Techonology Security Evaluation は、コンピュータ・セキュリティ認定の国際標準 (ISO/IEC 15408) です。Common Criteria は、カナダ、フランス、ドイツ、オランダ、イギリス、アメリカの政府によって開発されました。

Adaptive Server バージョン 15.0.1 は、2007 年 9 月に Common Criteria の検証を完了しています。評価済み設定は、セキュリティ・オプションとディレクトリ・サービス・オプションが設定された Adaptive Server バージョン 15.0.1 で構成されます。Adaptive Server のセキュリティ評価は、CCEVS (Common Criteria Evaluation and Validation Scheme) のプロセスとスキームに従って実施されました。Adaptive Server Enterprise の評価基準は、『Common Criteria for Information Technology Security Evaluation』(バージョン 2.3) と『International Interpretations effective』(2005 年 8 月付け) に記述されています。『Supplement for Installing Adaptive Server for Common Criteria Configuration』に従って設定することにより、Adaptive Server は、『Sybase® Adaptive Server Enterprise Security Target』(バージョン 1.5) に提示されているすべてのセキュリティ機能要件に適合します。

Adaptive Server は、次の 8 つのセキュリティ機能をサポートします。

- 暗号化サポート － Adaptive Server はカラム・レベルでのデータの透過的な暗号化をサポートしています。SQL 文と SQL 拡張機能により、安全なキー管理が提供されます。
- セキュリティ監査 － アクセス、認証の試行、管理者機能をチェックする監査メカニズムです。セキュリティ監査は、日付、時刻、責任者、イベントを記述するその他の詳細情報を監査証跡の中に記録します。
- ユーザ・データの保護 － 適用可能なデータベース・オブジェクト (データベース、テーブル、ビュー、ストアド・プロシージャ、暗号化キー) に対して任意のアクセス制御ポリシーを実装します。
- 識別と認証 － Adaptive Server は、基本となるオペレーティング・システムによるメカニズムに加え、独自の識別と認証メカニズムを備えています。
- セキュリティ管理 － ユーザとユーザに関連付けられている権限、アクセス・パーミッション、その他のセキュリティ機能 (監査証跡など) を管理する機能です。これらの機能は、ロール制限を含む任意アクセス制御ポリシー・ルールに基づいて制限されます。
- TOE Security Function (TSF) の保護 － Adpative Server は、コンテキストをユーザから分離し、オペレーティング・システムのメカニズムを使用して、Adaptive Server が使用するメモリとファイルに適切なアクセス設定が行われることを保証します。Adaptive Server は、セキュリティ・ポリシーの適用を保証するように設計された明確なインタフェースを使用してユーザと対話します。
- リソースの活用 － リソースを制限し、クエリやトランザクションによってサーバのリソースが独占されないようにします。
- Target of Evaluation (TOE) アクセス － 権限のある管理者は、特定のセッション数までログインを制限し、時間に基づいてアクセスを制限するログイン・トリガを構築できます。権限のある管理者は、ユーザ ID に基づいてアクセスを制限することもできます。

FIPS 140-2 検証済み暗号化モジュール

SSL は、インターネット上で取り扱われる、クレジット・カード番号、株式取引、銀行取引などの重要な情報を安全に転送するための標準です。Adaptive Server の SSL では、FIPS 140-2 レベル 1 評価の暗号化モジュールである Certicom Security Builder GSE が使用されています。詳細については、NIST Web サイト (http://csrc.nist.gov) で 2005 年 6 月 2 日付けの検証証明書 #542 を参照してください。

FIPS 140-2 認定の Certicom Security Builder GSE は、FIPS login password encryption 設定パラメータが有効な場合に、メモリやディスク上で転送されるログイン・パケットのログイン・パスワードを暗号化するためにも使用されます。

注意 SSL を使用して FIPS login password encryption パラメータを有効にするには、Security and Directory Services ライセンスが必要です。このパラメータが有効でない場合、OpenSSL セキュリティ・プロバイダを使用してログイン・パスワードの暗号化を実行します。

Adaptive Server 暗号化カラム機能は、対称鍵暗号法に依存しており、SSL と同じ FIPS 140-2 検証済み暗号化モジュールを使用します。『暗号化カラム・ユーザーズ・ガイド』を参照してください。

注意 Adaptive Server 暗号化カラム機能を使用するには、暗号化カラム・ライセンスが必要です。

第 2 章

Adaptive Server のセキュリティ管理について

セキュリティ管理の一般処理

「Adaptive Server を安全に管理するための主要タスクの実行」は、Adaptive Server のセキュリティ管理に必要な主要タスクの説明と、各タスクの実行方法についての指示が記載されているマニュアルを示します。

❖ Adaptive Server を安全に管理するための主要タスクの実行

1 監査機能を含む、Adaptive Server のインストール ― この作業には、インストールの準備、配布メディアからのファイルのロード、実際のインストール、必要な物理リソースの管理が含まれます。使用しているプラットフォームの『インストール・ガイド』および「第 8 章 監査」を参照してください。

2 安全な管理環境の設定 ― システム管理者とシステム・セキュリティ担当者を設定し、ログイン・プロファイルを作成し、パスワード・ポリシーとログイン・ポリシーを構築します。「第 3 章 Adaptive Server のログインおよびデータベース・ユーザの管理」を参照してください。

3 ログイン、データベース・ユーザ、および役割の設定 ― サーバにユーザ・ログインを追加してログイン・プロファイルを割り当てます。ユーザ定義の役割を作成し、役割の階層と役割の相互排他性を定義し、ログインに役割を割り当てます。ユーザをデータベースに追加します。「第 3 章 Adaptive Server のログインおよびデータベース・ユーザの管理」を参照してください。

4 ユーザ、グループ、役割のパーミッションの管理 － 特定の SQL コマンドの実行、特定のシステム・プロシージャの実行、データベース、テーブル、特定のテーブル・カラム、およびビューへのアクセスを実行するために必要なパーミッションの付与と取り消しを実行します。詳細なアクセス制御を実施するためのアクセス・ルールを作成します。「第 6 章 ユーザ・パーミッションの管理」を参照してください。

5 データベースの暗号化を設定し、テーブルの機密データを暗号化します。機密データの暗号化 － カラム・レベルでの暗号化を使用し、暗号化するデータ・カラムを決定し、一度でキー作成を行う操作を実行し、alter table を使用して初期データ暗号化を実行するために、Adaptive Server を設定します。『暗号化カラム・ユーザーズ・ガイド』を参照してください。

6 データ全体での整合性制御の設定 － 入力データを検証するために、検査制約、ドメイン・ルール、参照制約を追加します。『Transact-SQL ユーザーズ・ガイド』および『リファレンス・マニュアル：コマンド』を参照してください。

7 監査の設定と管理 － 監査対象を決定し、Adaptive Server の使用を監査します。また、監査証跡を使用して、システムへの侵入とリソースの不正使用を検出します。「第 8 章 監査」と、使用しているプラットフォームの『インストール・ガイド』と『設定ガイド』を参照してください。

8 高度な認証メカニズムとネットワーク・セキュリティを使用するためのインストール環境の設定 － LDAP、PAM、または Kerberos ベースのユーザ認証、暗号化によるデータ機密保持、データ整合性などのサービスを使用するようにサーバを設定します。「第 4 章 外部認証」と「第 7 章 データの機密保持」を参照してください。

セキュリティの設定に関する推奨事項

次に、ログインとセキュリティの関連について説明します。

- “sa” ログインの使用 － Adaptive Server をインストールする際に、システム管理者とシステム・セキュリティ担当者の役割を持つ “sa” という名前の単一のログインを設定します。このことは “sa” ログインがデータベースの処理に関して無制限の管理能力を持つことを意味します。

  “sa” ログインは、初期設定時にのみ使用してください。また、複数のユーザが “sa” アカウントを使用できるように設定するのではなく、各管理者に特定の役割を割り当てることによって、各ユーザの責任を明確にします。

- “sa” ログイン・パスワードの変更 − “sa” ログインの初期設定では、パスワードは “NULL” になっています。このパスワードは、インストール後すぐに alter login を使用して変更してください。

  警告！ Adaptive Server にログインするときは、isql の -P オプションを使用してパスワードを指定しないでください。他のユーザにパスワードを見られる可能性があります。

- 監査の有効化 − 監査は管理プロセスの早い段階で有効にしてください。このようにすれば、システム・セキュリティ担当者とシステム管理者によって実行される、権限が必要なコマンドの記録を取ることができます。この他に、データベースをダンプしたりロードしたりするオペレータなどの、特別な役割を持つユーザによって実行されたコマンドを監査することもできます。
- ログイン名の割り当て − Adaptive Server のログイン名には、オペレーティング・システムでのログイン名と同じ名前を割り当ててください。これにより、Adaptive Server へのログインが容易になり、サーバとオペレーティング・システムのログイン・アカウントの管理が簡単になります。また、Adaptive Server によって生成される監査データを、オペレーティング・システムの監査データと簡単に関連付けることができます。

セキュリティの設定例

表 2-1 に示すユーザに特別な役割を割り当てる場合を想定します。

表 2-1: 役割の割り当て対象ユーザ

名前	権限	オペレーティング・システムのログイン名
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 は、表 2-1 に示した役割の割り当てに基づいて Adaptive Server の安全な操作環境を設定するために使用する一連のコマンドを示します。オペレーティング・システムにログインしたら、初期設定されている “sa” アカウントを使用して次のコマンドを発行します。

表 2-2: セキュリティの設定に使用するコマンドの例

コマンド	結果
• isql -Usa	“sa” として Adaptive Server にログインする。sa_role と sso_role の両方がアクティブである。
• sp_audit “security”, “all”, “all”, “on”‘ • sp_audit “all”, “sa_role”, “all”, “on” • sp_audit “all”, “sso_role”, “all”, “on”‘	サーバ全体のセキュリティ関連イベントに対する監査オプションを設定する。また、sa_role または sso_role がアクティブなすべてのアクションの監査を設定する。
• sp_configure “auditing”, 1	監査を有効にする。
注意 監査証跡用のスレッショルド・プロシージャを設定し、sybsecurity でのトランザクション・ログの処理方法を決定してから、監査を有効化すること。「第 8 章 監査」 を参照。	
• create login	ログインとパスワードを追加する。
• grant role	役割を付与する。
• use sybsecurity • sp_changedbowner rsmith	システム・セキュリティ担当者の Rajnish をデータベース所有者にすることによって、監査データベース sybsecurity に対するアクセス権を付与する。Alan はシステム標準の役割を付与されない。
sp_locklogin sa,"lock"	他人が “sa” としてログインできないよう、“sa” ログインをロックする。各ユーザは、各自に設定された役割だけを使用できる。
注意 個々のユーザに sa_role と sso_role の各役割を付与し、これらの役割が正常に機能することを確認してから、“sa” ログインをロックすること。	

Adaptive Server のセキュリティ機能

次に、Adaptive Server の主要なセキュリティ機能を示します。

- 「識別と認証」(9 ページ) – 承認されたユーザだけがシステムにログインできるようにする。Adaptive Server は、パスワードベースのログイン認証の他に、Kerberos、LDAP、PAM による外部認証もサポートしている。
- 「任意アクセス制御」(10 ページ) – オブジェクトの所有者が、オブジェクトへのアクセスを制限できるようにするアクセス制御を提供する。通常は grant コマンドと revoke コマンドを使用する。この種の制御は、オブジェクトの所有者が自由に設定できる。
- 「役割の分担」(11 ページ) – 権限が付与された役割を複数の指定ユーザに割り当てて、指定ユーザだけが特定のタスクを実行できるようにする。Adaptive Server には、システム管理者やシステム・セキュリティ担当者などの「システム標準の役割」と呼ばれる、事前に定義された役割がある。また、システム・セキュリティ担当者が「ユーザ定義の役割」と呼ばれる追加の役割を定義できる。

- 「責任範囲の明確化のための監査」(12 ページ) − ログイン、ログアウト、サーバの起動操作、リモート・プロシージャ・コール、データベース・オブジェクトへのアクセス、特定ユーザによってまたは特定の役割をアクティブにして実行されたすべてのアクションなどのイベントを監査する機能。1 つのオプションを設定するだけで、サーバ全体にわたる一連のセキュリティ関連イベントを監査することもできる。
- 「データの機密保持」(13 ページ) − クライアント / サーバ間の通信に Kerberos や SSL による暗号化を使用して、データの機密性を保持する。カラム・レベルの暗号化では、データベースに保存されたデータの機密性を保持する。アクティブでないデータは、パスワードで保護されたデータベース・バックアップによって機密性を保持される。

識別と認証

Adaptive Server では、ログイン・アカウント名によってユーザをユニークに識別するために、サーバ・ユーザ ID (SUID) を使用します。この ID は各データベース内の特定のユーザ ID (UID) にリンクされています。アクセス制御では、SUID を持つユーザにオブジェクトへのアクセスを許可するかどうかを判断するときに、この ID が使用されます。認証では、ユーザが本人であることが確認されます。Adaptive Server では、内部認証メカニズムと外部認証メカニズムの両方を認証に使用できます。

識別と認証の詳細については、「第 3 章 Adaptive Server のログインおよびデータベース・ユーザの管理」を参照してください。また、「代理権限の使用」(188 ページ) と『システム管理ガイド 第 1 巻』の「第 7 章 リモート・サーバの管理」も参照してください。

外部認証

大規模な異機種アプリケーションでは、多くの場合、集中レポジトリでログインを認証することによってセキュリティを強化します。Adaptive Server では、次のようなさまざまな外部認証メソッドがサポートされています。

- Kerberos − Kerberos インフラストラクチャを含むエンタープライズ環境において、集中化された安全な認証メカニズムを提供します。KDC (Key Distribution Center) と呼ばれる信頼されたサード・パーティのサーバを使用して認証が行われ、クライアントとサーバの両方が検証されます。
- LDAP ユーザ認証 − LDAP (Lightweight Directory Access Protocol) は、ユーザのログイン名とパスワードに基づく集中化された認証メカニズムを提供します。
- PAM ユーザ認証 − PAM (Pluggable Authentication Module) は、管理操作およびランタイム・アプリケーション操作としてオペレーティング・システム・インタフェースを使用した、集中化された認証メカニズムを提供します。

これらの外部認証方式の詳細については、「第 4 章 外部認証」を参照してください。

リモート・サーバの管理

Adaptive Server 間でログインとユーザを管理する内部メカニズムについては、『システム管理ガイド 第 1 巻』の「第 7 章 リモート・サーバの管理」を参照してください。

任意アクセス制御

オブジェクト所有者は、そのオブジェクトに対するアクセス権を他のユーザに自由に付与できます。また、他のユーザにアクセス・パーミッションを付与できる権限を付与することもできます。Adaptive Server の任意アクセス制御は、grant コマンドによってユーザ、グループ、役割にさまざまな種類のパーミッションを与えることができるようにする機能です。パーミッションを取り消すには、revoke コマンドを使用します。grant コマンドと revoke コマンドは、指定のコマンドを実行したり、指定のテーブル、プロシージャ、ビュー、暗号化キー、カラムにアクセスしたりするためのパーミッションをユーザに与えます。

すべてのユーザがいつでもパーミッションなしで使用できるコマンドもあります。その他のコマンドは、システム管理者などの特定の役割のユーザだけが使用でき、譲渡することはできません。

権限の付与や取り消しが可能なコマンドにパーミッションを割り当てることができるかどうかは、各ユーザのステータス (システム管理者、システム・セキュリティ担当者、データベース所有者、データベース・オブジェクト所有者など) と、他のユーザにそのパーミッションを付与するオプション付きでパーミッションがユーザに付与されているかどうかによって決まります。

任意アクセス制御については、「第 6 章 ユーザ・パーミッションの管理」を参照してください。

ロー・レベル・アクセス制御

ロー・レベル・アクセス制御を使用すると、データをロー・レベルまで強力かつ柔軟に保護できます。管理者が個々のデータ要素の値に基づくアクセス・ルールを定義し、サーバがそれらのルールを透過的に適用します。管理者がアクセス・ルールを定義すると、アプリケーション、アドホック・クエリ、ストアド・プロシージャ、ビューなどで、影響を受けるデータのクエリが実行されるたびに、ルールが自動的に呼び出されます。

ルールベースのアクセス制御では、アプリケーションではなくサーバのセキュリティを強化するため、Adaptive Server のセキュリティ管理とアプリケーション開発プロセスの両方を簡略化できます。ロー・レベルのアクセス制御は、以下の機能を使用して実装できます。

- アクセス・ルール
- Application context facility
- ログイン・トリガ

- ドメイン整合性ルール

「ロー・レベル・アクセス制御の使用」(208 ページ) を参照してください。

役割の分担

Adaptive Server でサポートされる役割を使用すると、各ユーザの責任範囲を指定し、維持することができます。Adaptive Server には、システム管理者やシステム・セキュリティ担当者などのシステム標準の役割と、システム・セキュリティ担当者が作成するユーザ定義の役割があります。

役割とは、役割の割り付け対象者が自身のジョブを実行できるようにするための特権の集まりです。これらの役割により、システムの操作と管理作業を実行するユーザの責任が明確になります。そして、作業を監査し、どのユーザの作業かを明確にできます。

役割の階層

システム・セキュリティ担当者は、あるユーザがある役割を持つ場合、そのユーザはその階層内でそれよりも低い役割を自動的に持つ、というように役割の階層を定義できます。たとえば、役割 “chief_financial_officer” に、“financial_analyst” と “salary_administrator” の両方の役割が含まれるようにします。chief financial officer は、すべてのタスクを実行でき、salary administrator と financial analyst が参照可能なデータはすべて参照できます。

相互排他性

たとえば、次のような場合に、役割がメンバシップ・レベルとアクティブ化レベルで相互排他的になるように定義できます。次に例を示します。

- “payment_requestor” と “payment_approver” の両方の役割が同一ユーザに付与されないようにする場合。
- 1 人のユーザに “senior_auditor” と “equipment_buyer” の両方の役割が付与されていても、両方の役割を同時には有効にできないようにする場合。

システム標準の役割は、ユーザ定義の役割と同じく、役割階層内に定義することや、相互排他となるように定義することができます。たとえば、“super_user” という役割に、システム管理者、オペレータ、テクニカル・サポートの各役割が含まれるようにします。また、システム管理者とセキュリティ担当者の役割が、メンバシップに関して相互排他になるように、つまり、1 人のユーザに両方の役割を付与できないように定義できます。

「ユーザに対する役割の作成と割り当て」(145 ページ) を参照してください。

責任範囲の明確化のための監査

Adaptive Server には、総合的な監査システムがあります。監査システムは、次のものからなります。

- sybsecurity データベース
- 監査を管理するための設定パラメータ
- sp_audit はすべての監査オプションを設定します。
- sp_addauditrecord は、監査証跡にユーザ定義レコードを追加します。

監査機能のインストール時に、Adaptive Server が監査証跡に使用する監査テーブルの数を指定できます。複数のテーブルを使用して監査証跡を保管すると、オペレータの介入やレコードの損失のない、円滑に実行される監査システムを設定することができます。

システム・セキュリティ担当者は、監査システムを管理し、監査の開始と停止、監査オプションの設定、監査データの処理を実行できる唯一のユーザです。システム・セキュリティ担当者は、次のようなイベントの監査を設定できます。

- サーバ全体にわたるセキュリティ関連イベント
- データベース・オブジェクトの作成、削除、変更
- 特定ユーザが行ったすべてのアクション、または特定の役割をアクティブにしてユーザが行ったすべてのアクション
- データベース・アクセス権の付与または取り消し
- データのインポートまたはエクスポート
- ログインとログアウト
- 暗号化キーに関連するすべての作業

監査機能については、「第 8 章 監査」を参照してください。

データの機密保持

Adaptive server では、SSL (Secure Socket Layer) 標準や Kerberos を使用してクライアント・サーバ間の通信を暗号化することにより、データの機密性を保持できます。データベース内でカラム・レベルの暗号化を行い、オフライン・データのバックアップを暗号化することにより、データの機密性を保護することができます。dump コマンドと load database コマンドには、データベース・ダンプをパスワードで保護するための *password* パラメータが含まれています。

詳細については、以下を参照してください。

- SSL －「第 7 章 データの機密保持」
- Kerberos －「第 4 章 外部認証」
- 暗号化カラム －『暗号化カラム・ユーザーズ・ガイド』
- ダンプとロード －『リファレンス・マニュアル：コマンド』、『システム管理ガイド 第 2 巻』の「第 12 章 ユーザ・データベースのバックアップとリストア」

第 3 章

Adaptive Server のログインおよびデータベース・ユーザの管理

.

ログインおよびログイン・プロファイルの概要

ログインは、ユーザの名前とパスワードを定義して、Adaptive Server サーバへのアクセスを可能にします。create login が実行されると、Adaptive Server は master.dbo.syslogins にローを追加し、新しいユーザにユニークなシステム・ユーザ ID (suid) を割り当て、指定された属性情報を記録します。ユーザがログインするとき、Adaptive Server はそのユーザが指定した名前とパスワードを syslogins の中で検索します。password カラムは一方向アルゴリズムで暗号化されるので、解読することはできません。

ログイン・プロファイルとは、一連のログイン・アカウントに適用される属性の集合です。これらの属性は、プロファイルにバインドされている各ログインに対応したデフォルトの役割、ログイン・スクリプトなど、ログインの特性を定義します。ログイン・プロファイルではログイン・アカウントの属性が 1 か所で設定および管理されるため、システム・セキュリティ管理者にとっては時間の節約になります。

ログイン・アカウントの管理

Adaptive Server への新しいログイン・アカウントの追加、データベースへのユーザの追加、コマンドの使用およびデータベース・オブジェクトへのアクセスを行うための「パーミッション」の付与は、システム・セキュリティ担当者、システム管理者、データベース所有者で分担して行います。

表 3-1 には、ログイン・アカウントの作成および管理に使用するシステム・プロシージャがまとめられています。

表 3-1: Adaptive Server でのユーザの管理

タスク	必要な役割	コマンドまたはプロシージャ	データベース、グループ、または役割
ログイン・アカウントの作成	システム・セキュリティ担当者	create login	マスタ・データベース
ログイン・アカウントの変更	システム・セキュリティ担当者 例外は、ユーザ自身が自分のパスワードとフルネームを変更できる点です。	alter login	マスタ・データベース
ログイン・アカウントの削除	システム・セキュリティ担当者	drop login	マスタ・データベース
グループの作成	データベース所有者またはシステム管理者	sp_addgroup	ユーザ・データベース
役割の作成と割り当て	システム・セキュリティ担当者	create role、grant role	マスタ・データベース
データベースへのユーザの追加、グループの割り当て	データベース所有者またはシステム管理者	sp_adduser	ユーザ・データベース
その他のデータベース・ユーザに対するエイリアス・ユーザ	データベース所有者またはシステム管理者	sp_addalias	ユーザ・データベース
グループ、ユーザ、役割に対する、データベース・オブジェクトの作成パーミッションまたはアクセス・パーミッション、およびコマンドの実行パーミッションの付与	データベース所有者、システム管理者、システム・セキュリティ担当者、またはオブジェクト所有者	grant	ユーザ・データベース

ログイン・アカウントの作成

特定のサーバでのログイン・アカウントの作成とユーザ・パーミッションの管理は、次の手順で行われます。

1　システム・セキュリティ担当者が、新しいユーザのログイン・アカウントを作成します。

2　システム管理者またはデータベース所有者が、ユーザをデータベースに追加するかグループに割り当てます。

3　システム・セキュリティ担当者が、このユーザに特定の役割を付与します。

4　システム管理者、データベース所有者、またはオブジェクト所有者が、特定のコマンドとデータベース・オブジェクトに対するパーミッションを、ユーザまたはグループに付与します。

Adaptive Server に新しいログイン名を追加するには、create login を使用します。create login を実行できるのはシステム・セキュリティ担当者だけです。

完全な構文については、『リファレンス・マニュアル：コマンド』の「create login」を参照してください。

ログイン作成時に、syslogins の crdate カラムがそのときの日時に設定されます。

syslogins 内の suid カラムは、Adaptive Server 上の各ユーザをユニークに識別します。1 人のユーザの suid の値は、どのデータベースを使用する場合でも変わりません。Adaptive Server のインストール時に作成されるデフォルトの “sa” アカウントに割り当てられる suid の値は必ず 1 となります。他のユーザのサーバ・ユーザ ID は、create login が実行されるたびに Adaptive Server によって割り当てられる連続した整数値です。

パスワードの選択の詳細については、「パスワードの選択と作成」を参照してください。

次の文は、ユーザ “maryd” のアカウントを、パスワード “100cents”、デフォルト・データベース (master)、デフォルト言語 (us_english)、フルネームなしで設定します。

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

パスワードは 1 で始まるので、引用符が必要です。

この文が実行されると、“maryd” は Adaptive Server にログインできるようになります。このユーザは、master データベースへのアクセス権が明示的に付与されていなければ、master データベースでは “guest” ユーザとして扱われ、限定されたパーミッションが与えられます。

次の文は、ログイン・アカウント “omar_khayyam” とパスワード “rubaiyat” を設定して、“pubs2” をそのユーザのデフォルト・データベースにします。

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

最後のログインと非アクティブ・アカウントの管理

Adaptive Server はユーザ・アカウントのセキュリティ対策を、次の方法で行っています。

- 作成日を追跡する。
- アカウントに最後にログインした日時を記録する。
- 非アクティブになっているためロックしてもよいアカウントを特定する。
- アカウントがロックされるときには、その理由とロックを行ったユーザの ID を記録する。

stale period の定義

stale period はログイン・プロファイルの属性の 1 つで、ログイン・アカウントがロックされるまでに非アクティブであることができる期間を示します。ログイン・プロファイルの track lastlogin 属性が 0 に設定されておらず、そのログイン・アカウントには非アクティブであることを理由としたロックが適用されていないと、syslogins.lastlogindate フィールドと syslogins.pwdate フィールドでは、ログイン・プロセス時または sp_locklogin の実行時に非アクティブであったことの確認が行われます。

アクティブではないためにログイン・アカウントがロックされると、syslogins の locksuid、lockreason、lockdate の各フィールドは次のように設定されます。

lockreason の値	locksuid の値	アカウントの lockreason の説明
4	NULL	非アクティブであったため、アカウントが自動的にロックされました。

高可用性ソリューションが設定されている場合、syslogins.lastlogindate と syslogins.pwdate の両方がサーバで同期されます。特定のサーバでロックされているログイン・アカウントは、コンパニオン・サーバでもロックされます。

前回のログインの追跡

前回のログイン日時の追跡は、ログイン・プロファイルの track lastlogin 属性から設定できます。

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

非アクティブ・アカウントのロックの防止

exempt inactive lock 句を使用すると、ログイン・アカウントが非アクティブであってもロックされないようにすることができます。

次の文は、非アクティブによるロックが適用されないログイン・アカウント “user33” を作成します。

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

ログインの認証メカニズム

サポートされる認証メカニズムは、ASE、LDAP、PAM、KERBEROS、および ANY です。

ANY を使用すると、Adaptive Server は定義済みの外部認証メカニズムがあるかどうかを調べます。定義済みの外部認証メカニズムが検出された場合は、それを使用しますが、それ以外の場合は、ASE メカニズムを使用します。

ログイン・アカウントの変更

ログインの属性とその属性に対応する値を追加、削除、または変更するには、alter login を使用します。alter login を使用すると、次を実行できます。

- 自動的にアクティブ化された役割の追加または削除
- パスワードの変更
- ログイン・プロファイルの関連付けの変更
- フルネームの変更または追加
- パスワードの有効期限と最小パスワード長の指定
- 最大失敗回数の指定
- 認証メカニズムの指定
- デフォルト言語およびデフォルト・データベースの指定
- ログイン・スクリプトの起動
- 非アクティブなログイン・アカウントの適用除外

システム管理者は、alter login を使用すると、パスワードの長さと有効期間を設定したり、ログイン試行の失敗回数を制限したり、属性を削除したり、ログイン時にログイン･スクリプトを自動的に実行するように指定したりできます。

alter login を実行してデフォルト・データベースを変更すると、ユーザは次回ログインするときに新しいデフォルト・データベースに接続されます。ただし、alter login を実行しても、そのデータベースに対するアクセス権がユーザに自動的に与えられることはありません。データベース所有者が sp_adduser または sp_addalias を使用してアクセス権を設定するか、guest ユーザを使用してアクセスできるように設定しなければ、ユーザのデフォルト・データベースが変更されても、そのユーザは master データベースに接続されます。

次の例では、anna というログイン・アカウントのデフォルト・データベースを pubs2 に変更します。

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

次の例では、claire のデフォルト言語をフランス語に変更します。

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

ログイン・アカウントの削除

drop login コマンドを実行すると、そのユーザのエントリが master.dbo.syslogins から削除され、Adaptive Server のユーザ・ログインが削除されます。

通常、データベース内でユーザであるログインを削除することはできません。また、ユーザがそのデータベース内でオブジェクトを所有している場合、または、他のユーザにオブジェクトのパーミッションを付与している場合は、データベースからユーザを削除できません。

次のログイン・アカウントが作成されるときに、削除されたログイン・アカウントのサーバ・ユーザ ID (suid) が再使用されることがあります。これは、削除されたログインが syslogins 内で最上位の suid を保持している場合にだけ起こります。ただし、drop login の実行が監査されていない場合は、アカウントの信頼性が低下する可能性があります。

最後に残ったシステム・セキュリティ担当者またはシステム管理者のログイン・アカウントは、削除できません。

with override 句は、ログイン参照をチェックできない使用不可のデータベースがある場合でも、ログインを削除します。

次の例では、mikeb ログイン・アカウントと rchin ログイン・アカウントを削除します。

```
drop login mikeb, rchin
```

drop login 構文の詳細については、『リファレンス・マニュアル：コマンド』を参照してください。

パスワードの選択と作成

システム・セキュリティ担当者は、ユーザを Adaptive Server へのログインに追加するときに、create login を使用して各ユーザにパスワードを割り当てます。alter login 文を使用すると、ユーザがいつでも自分のパスワードを変更できます。「パスワードの変更」(77 ページ) を参照してください。

パスワードを作成するときは、次の規則に従います。

- 誕生日や住所など、個人の生活に関係する言葉や数字を使用しない。
- ペットや家族などの名前を使用しない。
- 辞書にある言葉や、単語のスペルを逆にしたものを使用しない。

最も推察しにくいパスワードは、大文字、小文字、数字を組み合わせたものです。自分のパスワードは決して他人に教えたり、他人の見ている前で紙に書き留めたりしないでください。

次にパスワードの規則を示します。

- パスワードの長さは、6 文字以上でなければならない。Sybase では、8 文字よりも長いパスワードをおすすめします。
- 印刷可能な文字、数字、または記号で構成されている。
- 次の場合は、create login で指定するときにパスワードを引用符で囲む。
  - A ～ Z、a ～ z、0 ～ 9、_、#、有効な 1 バイトまたはマルチバイトのアルファベット文字以外の文字を含む場合、またはアクセント付きのアルファベット文字を含む場合
  - 0 ～ 9 の数字で始まる場合。

「複雑なパスワード・チェック」(27 ページ) を参照してください。

ログインを試行できる最大回数の設定と変更

ログインの最大試行回数を設定すると、当て推量や辞書を使ったパスワードの推測を防止できます。システム・セキュリティ担当者は、ログインが連続して何回試行されたらログインや役割を自動的にロックするかを指定できます。ログイン試行回数の最大数は、サーバワイドまたはログインや役割ごとに設定できます。個々のログインや役割の設定は、サーバワイドの設定よりも優先されます。

ログインの失敗回数は、master..syslogins の logincount カラムに格納されます。正常にログインすると、失敗したログインの数が 0 にリセットされます。

❖ **サーバ全体での *maximum failed logins* の設定**

デフォルトでは、maximum failed logins はオフになっており、このチェックはパスワードに適用されません。sp_passwordpolicy は、ログインや役割に対するログイン失敗の最大回数をサーバワイドで設定するときに使用します。

- 許可されるログイン失敗回数を設定するには、次のように入力します。

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

『リファレンス・マニュアル：プロシージャ』の「sp_passwordpolicy」を参照してください。

❖ **特定のログインの *maximum failed logins* の設定**

- 特定のログインの作成時にログイン試行最大失敗回数を設定するには、create login を使用します。

この例では、パスワードが “Djdiek3” である新しいログイン “joe” を作成します。このとき、ログイン “joe” に対する最大ログイン試行回数を 3 に設定します。

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

『リファレンス・マニュアル：コマンド』の「create login」を参照してください。

❖ **特定の役割の *maximum failed logins* の設定**

- 特定の役割の作成時に maximum failed logins を設定するには、create role を使用します。

次の例では、パスワードが “temp244” である役割 “intern_role” を作成します。このとき、“inter_role” に対する maximum failed logins を 20 に設定します。

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

『リファレンス・マニュアル：コマンド』の「create role」を参照してください。

❖ **特定のログインの *maximum failed logins* の変更**

- 既存のログインに対する最大ログイン失敗回数を設定または変更するには、alter login を使用します。

ログイン “joe” の maximum failed logins を 40 に変更します。

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

❖ **特定の役割の *maximum failed logins* の変更**

- 既存の役割に対する maximum failed logins を設定または変更するには、alter role を使用します。

次の例では、“physician_role” に対する maximum failed logins を 5 に変更します。

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

次の例では、すべての役割に対する maximum failed logins を無効にする設定を削除します。

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

maximum failed logins を使用するための構文と規則の詳細については、『リファレンス・マニュアル：コマンド』の「alter role」を参照してください。

パスワードが失われた場合のログイン

dataserver -p*login_name* パラメータを使用して、サーバの起動時にシステム・セキュリティ担当者またはシステム管理者の名前を指定します。これによって、失われたパスワードをリカバリする方法がない場合に、これらのアカウントの新しいパスワードを設定できます。

-p パラメータを使用して起動すると、Adaptive Server は、ランダムなパスワードを生成、表示、暗号化してから、そのアカウントの新しいパスワードとして master..syslogins に保存します。

dataserver –p を使用して、sa_role と sso_role のパスワードを再設定できます。これらの役割のパスワードが失われた場合は dataserver -p を使用しますが、役割のパスワードをアクティブにする必要があります。

たとえば、次のように入力してサーバを起動したとします。

```
dataserver -psa_role
```

Adaptive Server は次のメッセージを表示します。

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

sa_role のパスワードがない場合に -psa_role を使用して起動すると、Adaptive Server はエラー・ログにエラー・メッセージを出力します。

サーバの再起動時に、ログインまたは役割のパスワードを変更することを強くおすすめします。

パスワード情報の表示

この項では、ログインと役割のパスワード情報の表示方法について説明します。

❖ 特定のログインのパスワード情報の表示

- 特定のログインのログイン設定とパスワード設定を表示するには、sp_displaylogin を使用します。

次の例では、ログイン・プロファイルにバインドされている joe というログインについての情報を表示します。

```
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
Login Profile: emp_lp
```

次の例では、ログイン・プロファイルにバインドされていない joe というログインについての情報を表示します。

```
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
```

『リファレンス・マニュアル：プロシージャ』の「sp_displaylogin」を参照してください。

❖ 特定の役割のパスワード情報の表示

- 役割のログイン設定とパスワード設定を表示するには、sp_displayroles を使用します。

この例では、役割 physician_role についての情報が表示されます。

```
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
```

『リファレンス・マニュアル：プロシージャ』の「sp_displayroles」を参照してください。

パスワードが 1 文字以上あるかどうかの検査

システム・セキュリティ担当者は、サーバワイドの設定パラメータ check password for digit を使用して、パスワードが 1 文字以上あることをチェックするようにサーバに指示することができます。このパラメータを設定しても、既存のパスワードに影響を与えることはありません。デフォルトでは、1 文字以上あるかどうかの検査は行われません。

次の例では、パスワードの検査機能をアクティブにします。

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

この例では、パスワードの検査機能を非アクティブにします。

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

『リファレンス・マニュアル：プロシージャ』の「sp_configure」を参照してください。

最小パスワード長の設定と変更

現在ではパスワードの最小長を設定できるようになっているので、たとえば、4 桁の個人識別番号 (PIN) や、NULL パスワードによる匿名ログインの使用など、ニーズに応じてパスワードをカスタマイズできます。

注意　Adaptive Server は、minimum password length (最小パスワード長) にデフォルト値の 6 を使用します。このパラメータを 6 以上の値に設定することをおすすめします。

システム・セキュリティ担当者は、以下のものを指定できます。

- システム全体にわたって強制される minimum password length
- ログインごと、または役割ごとの minimum password length

ログインごとの値または役割ごとの値は、サーバワイドの値よりも優先されます。minimum password length の設定は、値を設定した後に作成した新しいパスワードにのみ反映されます。

❖ **特定のログインに対する *minimum password length* の設定**

- ログインを作成するときに、そのログインに対する最小パスワード長を設定するには、create login を使用します。

この例では、パスワードが “Djdiek3” である新しいログイン “joe” を作成します。このとき、“joe” の minimum password length を 8 に設定します。

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

『リファレンス・マニュアル：コマンド』の「create login」を参照してください。

❖ **特定の役割に対する *minimum password length* の設定**

- create role を使用して役割を作成するときに、その役割の minimum password length を設定できます。

次の例では、パスワードが “temp244” である役割 “intern_role” を作成します。このとき、“intern_role” の minimum password length を 0 に設定します。

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

元のパスワードは 7 文字ですが、minimum password length が 0 に設定されているため、変更するパスワードの長さの制限はありません。

『リファレンス・マニュアル：コマンド』の「create role」を参照してください。

❖ **特定のログインに対する *minimum password length* の変更**

- 既存のログインに対する minimum password length を設定または変更するには、alter login を使用します。

次の例では、ログイン “joe” の minimum password length を 8 文字に変更します。

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

『リファレンス・マニュアル：コマンド』の「alter login」を参照してください。

❖ **特定の役割に対する *minimum password length* の変更**

- 既存の役割の minimum password length を設定または変更するには、alter role を使用します。

次の例では、既存の役割である “physician_role” の最小パスワード長を 5 文字に設定します。

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

次の例では、すべての役割の minimum password length を無効にします。

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

『リファレンス・マニュアル：コマンド』の「alter role」を参照してください。

❖ **特定のログインの *minimum password length* の解除**

- 既存のログインの最小パスワード長を解除するには、alter login を使用します。

次の例では、ログイン joe についての minimum password length の制限を削除します。

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

『リファレンス・マニュアル：コマンド』の「alter login」を参照してください。

複雑なパスワード・チェック

ストアド・プロシージャ・インタフェースで、複雑なパスワード・チェックをサポートする次のオプションを使用できます。その値は、master.dbo.sysattributes テーブルに格納されます。

個々のオプションをオフにするには、次のように入力します。

```
sp_passwordpolicy 'clear', option
```

すべてのパスワードのポリシー・オプションをオフにするには、次のように入力します。

```
sp_passwordpolicy 'clear'
```

複雑なログイン・パスワード・チェックは、役割のパスワードまでにも及びます。「役割パスワードを確認するログイン・パスワード・ポリシー」(164 ページ) を参照してください。

sp_passwordpolicy 構文の詳細については、『リファレンス・マニュアル：プロシージャ』を参照してください。

単純なパスワードの禁止

disallow simple password では、パスワードにログイン名が部分文字列として含まれていないかチェックします。次のように設定できます。

- 0 − (デフォルト) このオプションをオフにし、単純なパスワードを許可する。
- 1 − このオプションをオンにし、単純なパスワードを禁止する。

このオプションを設定するには、次のように入力します。

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

カスタムの複雑なパスワード・チェック

Adaptive Server では、sp_extrapwdchecks と sp_cleanpwdchecks を使用してパスワード・チェックのルールをカスタム設定できます。

これらのストアド・プロシージャは、master データベースで定義および配置されており、Adaptive Server による複雑なパスワード・チェック中に自動的に呼び出され、この時点でログインがそれぞれ破棄されます。これらのカスタム・ストアド・プロシージャの作成例については、「カスタムのパスワード・チェックの有効化」(32 ページ) を参照してください。

パスワードの文字数の指定

これらの sp_passwordpolicy パラメータを使用して、パスワード中の最小文字数 (桁数や大文字と小文字など) を指定します。

- min digits in password − パスワードの最小桁数。デフォルトでは無効です。有効な値は次のとおりです。
  - 0 ～ 16 − パスワードに必要な数字の最小文字数。
  - -1 − パスワードに数値を含めることはできない。
- min alpha in password − パスワードで使用できるアルファベットの最小文字数。この値は、大文字と小文字の最小数を合わせた文字数以上の値にする必要があります。デフォルトでは無効です。有効な値は次のとおりです。
  - 0 ～ 16 − パスワードのアルファベットの最小文字数。
  - -1 − パスワードに特殊文字を含めることはできない。
- min special char in password − パスワードの特殊文字の最小文字数。有効な値は次のとおりです。
  - 0 ～ 16 − パスワードの特殊文字の最小文字数。
  - -1 − パスワードに特殊文字を含めることはできない。
- min upper char in password − パスワードの大文字の最小文字数。デフォルトでは無効です。有効な値は次のとおりです。
  - 0 ～ 16 − パスワードに必要な大文字の文字数。
  - -1 − パスワードに大文字を含めることはできない。
- min lower char in password − パスワードの小文字の最小文字数。有効な値は次のとおりです。
  - 0 ～ 16 − パスワードに必要な大文字の文字数。
  - -1 − パスワードに大文字を含めることはできない。

- minimum password length － 最小パスワード長。最小パスワード長は 0 ～ 30 の範囲で設定できます。指定する値は、他の最小要件をすべて組み合わせた長さ以上にする必要があります。たとえば、次のように設定している場合は、minimum password length を 10 以上に設定する必要があります。
  - minimum digits in password を 3 に設定
  - minimum special characters in password を 2 に設定
  - minimum uppercase characters in password を 2 に設定
  - minimum lowercase characters in password を 3 に設定
- password expiration － 期限が切れるまでの、パスワードが存在できる日数。この値はグローバルな単位で指定します。デフォルトでは無効です。有効な値は次のとおりです。
  - 0 － パスワードの期限は切れない。
  - 1 ～ 32767 － 期限が切れるまでの、パスワードが存在できる日数。
- password exp warn interval － パスワードの期限が切れるまで、パスワード有効期限の警告メッセージを表示する間隔 (日数)。これらのメッセージは、パスワードが変更されるか、期限が切れるまで、成功したすべてのログインで表示されます。この値は、パスワード有効期限以下の値にする必要があります。デフォルトでは無効です。

  有効値は 0 ～ 365 です。
- maximum failed logins － ログインがロックされるまで実行できる、ログイン失敗の最大回数。この値はグローバルに指定します。デフォルトでは無効です。有効な値は次のとおりです。
  - 0 － ログイン失敗回数に関係なく、ログインはロックされない。
  - 1 ～ 32767 － ログインがロックされるまでに許可されるログイン失敗回数。
- expire login では、システム・セキュリティ担当者がログインを作成またはリセットすると、ログインのステータスを期限切れに変更します。ログインは、初回ログイン時にパスワードを変更する必要があります。デフォルトでは無効です。有効な値は次のとおりです。
  - 0 － 新しいログインまたはリセットされたログインには期限を設定しない。
  - 1 － 新しいログインまたはリセットされたログインの期限が切れた場合は、初回ログイン時にパスワードをリセットする必要がある。

『リファレンス・マニュアル：プロシージャ』の「sp_passwordpolicy」を参照してください。

複雑なパスワード・オプションの相互チェック

複雑なパスワード･オプションには､次の対話的な機能を持つものがあります。

- minimum password length には、min digits in password、min alpha in password、min special characters in password の合計以上の値を設定する。
- min alpha in password には､min upper char in password および min lower char in password の合計以上の値を設定する。
- systemwide password expiration には、password exp warn interval よりも大きい値を設定する。

上記の相互チェックを行うために、Adaptive Server では、値が -1 の複雑なパスワード・オプションが検出された場合、この値は 0 と解釈されます。オプションが設定されていない場合も、そのオプションの値は 0 と解釈されます。

Adaptive Server では、相互チェックに合格しない新しい複雑なパスワード・オプションそれぞれについて警告を表示します。ただし、オプションの設定は成功します。

複雑なパスワード・チェックの設定

表 3-2: 複雑なパスワード・チェック

Adaptive Server 認証のパスワード・チェックとポリシー	sp_configure を使用して指定される設定パラメータ	sp_passwordpolicy を使用して指定される複雑なパスワード・オプション	alter login を使用して指定されるログイン単位の上書き
パスワードの有効期限	system-wide password expiration	system-wide password expiration	password expiration
パスワードの数字の文字数	check password for digit	min digits in password	該当なし
パスワードのアルファベット文字数	該当なし	min alpha in password	該当なし
パスワードの長さ	minimum password length	minimum password length	min passwd length
ロックされるまでのログイン失敗回数	maximum failed logins	maximum failed logins	max failed attempts
単純なパスワードの禁止	該当なし	disallow simple passwords	該当なし
パスワードの特殊文字数	該当なし	min special char in password	該当なし
パスワードの大文字数	該当なし	min upper char in password	該当なし
パスワードの小文字数	該当なし	min lower char in password	該当なし
パスワード有効期限の警告間隔	該当なし	password exp warn interval	該当なし
初回ログイン時のパスワードのリセット	該当なし	expire login	該当なし
カスタムの複雑なパスワード・チェック	該当なし	該当なし	該当なし

複雑なパスワード・オプションは次のレベルで設定できます。

- ログイン・レベル。create login または alter login を使用する。
- グローバル・レベル。新しい sp_passwordpolicy または sp_configure を使用する。

グローバル単位およびログイン単位で、古いパラメータと新しいパラメータを使用してパスワード設定オプションを設定できるため、パスワード・オプションが適用される優先順位は重要です。

パスワード・オプションを適用すると、優先順位は次のようになります。

1　既存のログイン単位のパラメータ

2　複雑なパスワード・オプション

3　既存のグローバル・パスワード・オプション

例

例 1　新しいログインを作成し、“johnd” の最小パスワード長に 6 を設定します。

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

ログイン “johnd” に対する上記のグローバル・オプションによって、ログイン “johnd” に対する 2 つの最小パスワード長要件が作成され、パスワードの桁数の制限についても設定されます。

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

次に、ログイン “johnd” のパスワードを次のように変更します。

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

Adaptive Server では、次の順序でパスワードをチェックします。

1　ログイン単位の既存のオプションのチェック：パスワードの最小長は 6 より大きい値にする必要があります。これには該当するため、チェックは合格です。

2　新しいオプション：パスワードの最小桁数は 2 より大きい値にする必要があります。これには該当するため、チェックは合格です。

3　既存のグローバル・オプション：ログイン “johnd” についてはログイン単位のチェックが既に行われているため、この例で指定されている最小パスワード長はチェックされません。

4　パスワードの桁のチェック・オプションは、最小桁数がオンで、値が 2 に設定されているときに既にチェックされているため、不要です。

Adaptive Server が指定された順序をチェックし、ログイン “johnd” の新しいパスワードがこれらのチェックに合格すると、パスワードの変更は成功します。

例 2 ユーザ “johnd” について次のように入力すると、Adaptive Server は最初にログイン単位の既存のオプションをチェックし、最小パスワード長が 6 に設定されることを確認します。しかしユーザは、4 文字のみを使用するパスワードを変更しようとしました。

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

この場合チェックは失敗し、Adaptive Server はエラー・メッセージを示します。1 つの複雑なパスワード・チェックが失敗すると、それ以外のオプションはチェックされません。

例 3 パスワード設定オプションを指定して新しいログインを作成し、ログイン johnd の最小パスワード長を 4 に設定します。

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

これはログイン単位の既存のオプションです。その後次のオプションを追加すると、パスワードの最小桁数を 1 に設定する必要があるグローバル要件が作成されます。

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

次に、ログイン johnd のパスワードを次のように変更します。

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

Adaptive Server では、次の順序でチェックを実行します。

1 ログイン単位の既存のオプションのチェック：新しいパスワードの最小パスワード長は 4 です。パスワード “abcde” は 4 文字を超えているため、このチェックは合格です。

2 新しいグローバル要件のチェック：パスワードの最小桁数はグローバル単位で 1 に設定されています。このチェックは失敗します。

Adaptive Server はパスワードを変更せずに、エラー・メッセージを示します。

パスワードを変更するには、すべてのチェックに合格する必要があります。

カスタムのパスワード・チェックの有効化

Adaptive Server では、システム・セキュリティ担当者が、カスタムのパスワード・チェックを有効にするユーザ定義のストアド・プロシージャを作成できます。

たとえば、パスワード履歴のチェックを実装するには、次のように入力して、パスワードの履歴を保存するための新しいユーザ・テーブルを作成します。

```
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
```

このユーザ定義のストアド・プロシージャ (sp_extrapwdchecks) は、新しいパスワードを pwdhistory テーブルに暗号化フォームで保存することを指定する場合に呼び出すことができます。

```
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
```

sp_addmessage を使用して、ユーザ定義のメッセージ 22001 を追加します。raiserror 22001 は、カスタムの複雑なパスワード・チェックのエラーが発生したことを示します。

次のユーザ定義のストアド・プロシージャ (sp_cleanpwdchecks) は、sp_extrapwdchecks を使用してパスワード履歴をクリーンアップするために使用できます。

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

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

上記の 2 つのプロシージャが定義され、master データベースにインストールされると、これらのパラメータは複雑なパスワード・チェック中に動的に呼び出されます。

パスワードのログインと役割の有効期間の設定

システム管理者とシステム・セキュリティ担当者は次のことができます。

使用	目的
create login	作成時にログイン・パスワードの有効期間を指定する。
alter login	ログイン・パスワードの有効期間を変更する。
create role	作成時に役割のパスワードの有効期間を指定する (create role を発行できるのは、システム・セキュリティ担当者のみです)。
alter role	役割のパスワードの有効期間を変更する (alter role を発行できるのは、システム・セキュリティ担当者のみです)。

ログインと役割に対して設定するパスワードの有効期間には、次の規則が適用されます。

- ログイン・アカウントごと、または役割ごとに割り当てたパスワード有効期間は、システム全体にわたるパスワード有効期間の値よりも優先される。これによって、システム・セキュリティ担当者のパスワードなどの機密性の高いアカウントまたは役割には比較的短いパスワード有効期間を指定し、匿名ログインなどの機密性の低いアカウントには比較的長い有効期間を指定できる。

- パスワードの有効期間が切れているログインまたは役割は、直接アクティブにはならない。
- パスワードは、password expiration interval によって指定された日数が過ぎた後、パスワードを最後に変更した日に有効期限が切れます。

コマンドおよびシステム・プロシージャの構文と規則の詳細については、適切な『リファレンス・マニュアル』を参照してください。

12.x より前のパスワードにはパスワード有効期間が無効

Adaptive Server 12.x より前のバージョンでは、役割はパスワード有効期間の影響を受けていませんでした。Adaptive Server 12.x 以降では、既存のユーザ定義の役割のパスワードに対するパスワード有効期間はアクティブにはなりません。

パスワードによる保護の迂回

自動ログイン・システムでは、パスワードによる保護を回避する必要がある場合があります。パスワードを入力しなくても他の役割にアクセスできる役割を作成することができます。

特定のユーザについてはパスワードによる保護を行わないようにする場合は、パスワードで保護されている役割を別の役割に付与し、このパスワードで保護された役割を 1 人または複数のユーザに付与します。この役割をアクティブにすると、パスワードを入力しなくても、パスワードで保護されている役割が自動的にアクティブ化されます。

次に例を示します。

Jane は ABS Inc. のシステム・セキュリティ担当者で、自動ログイン・システムを使用しています。Jane は次の役割を作成します。

- 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 は financial_assistant と accounts_officer の役割を chief_financial_officer の役割に付与します。

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

次に、chief_financial_officer の役割を Bob に付与します。

```
grant role chief_financial_officer to bob
```

Bob は Adaptive Server にログインし、chief_financial_officer の役割をアクティブにします。

```
set role chief_financial_officer on
```

financial_assistant と accounts_officer の役割は、Bob がパスワードを入力しなくても自動的にアクティブになります。これで Bob は、パスワードを入力しなくても、役割 financial_assistant と accounts_officer の管理下にあるすべてのデータにアクセスできます。

新規ログインのパスワードの有効期間の作成

新しいログインに対してパスワードの有効期間を設定するには、create login を使用します。

この例では、パスワードが “Djdiek3” である新しいログイン “joe” を作成します。このとき、“joe” のパスワードの有効期間を 2 日間に設定します。

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

“joe” のパスワードは、ログイン・アカウントが作成された日から 30 日後、またはパスワードを最後に変更した日から 30 日後に有効期限が切れます。

『リファレンス・マニュアル：プロシージャ』の「create login」を参照してください。

新規役割のパスワードの有効期間の作成

新しい役割に対してパスワードの有効期間を設定するには、create role を使用します。

次の例では、パスワードが “temp244” である新しい役割 intern_role を作成します。このとき、intern_role のパスワードの有効期間を 7 日間に設定します。

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

intern_role のパスワードは、この役割を作成した日から 7 日後、またはパスワードを最後に変更した日から 2 日後に有効期限が切れます。

『リファレンス・マニュアル：コマンド』の「create role」を参照してください。

パスワードの作成日の刻印

パスワードには、サーバがアップグレードされた日が「作成日」として刻印されます。ログイン・パスワードの作成日は、syslogins の pwdate カラムに格納されます。役割のパスワードの作成日は、syssrvroles の pwdate カラムに格納されます。

ログインや役割に設定されているパスワード有効期間の変更または削除

既存のログインに設定されているパスワード有効期間を変更または削除したり、有効期間が設定されていないログインに有効期間を設定したりするには、alter login を使用します。alter login は、ログイン・パスワードだけに影響し、役割のパスワードには影響しません。

次の例では、ログイン “joe” のパスワード有効期間を 5 日間に変更します。

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

パスワードは、パスワード有効期限を過ぎた日から 30 日後に期限切れになります。

『リファレンス・マニュアル：コマンド』の「alter login」を参照してください。

ネットワーク上でのログイン・パスワードの保護

クライアントからサーバへパスワードを安全に転送するために、Adaptive Server は RSA パブリック・キー暗号化アルゴリズムを使用して非対称暗号化の使用を可能にしています。Adaptive Server は非対称キーのペアを生成して、ログイン・プロトコルを使用するクライアントにそのパブリック・キーを送信する。たとえば、クライアントはパブリック・キーを使用してユーザのログイン・パスワードを暗号化してからサーバに送信する。サーバはプライベート・キーを使用してパスワードを解読し、接続しようとしているクライアントの認証を開始する。

Adaptive Server がクライアントにログイン・プロトコルの使用を要求するように設定できます。Adaptive Server の設定パラメータ net password encryption reqd を設定して、ユーザ名とパスワードに基づくすべての認証で RSA 非対称暗号化の使用を要求できる。『システム管理ガイド 第 1 巻』の「第 5 章 設定パラメータ」の「net password encryption required」を参照してください。

非対称キー・ペアの生成

Adaptive Server が新しいキー・ペアを生成するのは次のような場合です。

- サーバ起動時
- 自動的に 24 時間間隔で、Adaptive Server ハウスキーピング・メカニズムによって
- sso_role を持つ管理者がキー・ペアの再生成を要求したとき

キー・ペアはメモリに保管されます。キー・ペアが再生成されるとエラー・ログと監査証跡にメッセージが記録されます。

次のコマンドを使用すると、いつでもキー・ペアを生成できます。

```
sp_passwordpolicy "regenerate keypair"
```

注意 システムの負荷状態によっては、このコマンドを実行してからキー・ペアが実際に生成されるまでしばらく時間がかかる場合があります。これはハウスキーピング・タスクの優先度が低いため、優先度の高いタスクが終了するのを待つことになる場合があるからです。

キー・ペアの生成時刻を指定するには、次のコマンドを使用します。

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

ここで、*datetime* はキー・ペアを再生成する日付と時刻です。

たとえば、日時文字列として “Jan 16, 2007 11:00PM” を指定すると、その時刻にキー・ペアが生成されます。日時文字列には “4:07AM” のように時刻のみを指定することもできます。時刻のみを指定すると 24 時間以内の該当する時刻にキー・ペアが生成されます。

sp_passwordpolicy を使用すると、キー・ペアを再生成する頻度に加え、キー・ペアの生成に失敗したときの Adaptive Server の動作も設定できます。

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

『リファレンス・マニュアル：システム・プロシージャ』の「sp_passwordpolicy」を参照してください。

サーバ・オプション "net password encryption"

Adaptive Server はリモート・プロシージャ・コール (RPC) を確立するときにクライアントとしても機能します。

リモート・サーバに接続するとき、Adaptive Server は net password encryption オプションを使用してパスワードの暗号化を使用するかどうかを判断します。

このサーバ・オプションが true に設定されていると Adaptive Server は RSA または Sybase 独自のアルゴリズムを使用します。net password encryption を有効にするには次のコマンドを使用します。

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

設定は master..sysservers に保管され、サーバ・オプションの値は sp_helpserver ストアド・プロシージャを使用して表示できます。

sp_addserver を使用して追加された新しいサーバでの net password encryption のデフォルト値は true になります。アップグレード時に、Adaptive Server は ASEnterprise クラス値を持つ sysservers エントリの net password encryption を “true” に設定します。他のサーバ・クラスは変更されません。これによって、Adaptive Servers 間でのパスワード・セキュリティが向上します。

注意 サーバへの接続の確立に問題が発生した場合、管理者は net password encryption を false にリセットすることもできます。ただし、false に設定した場合、パスワードはネットワーク上でのクリア・テキストとして送信されます。

旧バージョンとの互換性

- Sybase ではネットワーク上でパスワードを保護するために RSA アルゴリズムを使用するようおすすめしています。
- RSA アルゴリズムを使用するには、Adaptive Server バージョン 15.0.2 と新しい Connectivity SDK クライアント (バージョン 15.0 ESD#7 以降) が必要です。Sybase では net password encryption reqd 設定パラメータと net password encryption サーバ・オプションを用意することによって 15.0.2 より古いバージョンでの設定と同じ設定を使用できるようにして、旧バージョンのクライアントやサーバとの互換性を維持しています。
- RSA アルゴリズムをサポートしない古いクライアントではそのプロパティに、バージョン 12.0 以前から使用されてきた Sybase 独自のアルゴリズムによるパスワード暗号化を設定できます。そうすれば、Adaptive Server は Sybase 独自のアルゴリズムを使用します。
- RSA アルゴリズムも Sybase 独自のアルゴリズムもサポートする新しいクライアントでは、両方のアルゴリズムをプロパティに設定します。そのようなクライアントと通信するとき Adaptive Server 15.0.2 は RSA 暗号化を使用します。15.0.2 より古い Adaptive Server は Sybase 独自のアルゴリズムを使用します。

ディスクとメモリに保管されているログイン・パスワードの保護

Adaptive Server がクライアント接続の認証で使用するログイン・パスワードは SHA-256 ハッシュ・ダイジェストとしてディスクに安全に保管されています。SHA-256 アルゴリズムは一方通行の暗号化アルゴリズムです。生成されたダイジェストは解読不能なので、ディスクへ保管しても安全です。ユーザ接続の認証では、クライアントから送られてきたパスワードに SHA-256 アルゴリズムが適用され、その結果がディスクに保管されている値と比較されます。

ディスクに保存されたログイン・パスワードに対する辞書ベースの攻撃を防ぐために、SHA-256 アルゴリズムを適用する前にパスワードにソルトが混入されます。ソルトは SHA-256 ハッシュとともに保管され、ログイン認証時に使用されます。

以前のバージョンへのダウングレードがないことが確かな場合は、SHA-256 のみを使用することをおすすめします。そうすると 15.0.2 より前のバージョンへのダウングレードが必要になった場合、管理者がユーザ・ログイン・パスワードのロック解除に介入しなければならないことを考慮する必要があります。

パスワード文字セットの考慮事項

暗号化されているパスワードおよびその他の機密データを認証のために復号化したりハッシュ値を比較したりするときに、その結果を正確に解釈するためにはクリア・テキストの文字セットを決定する必要があります。

たとえば、クライアントが isql を使用して Adaptive Server に接続し、新しいパスワードを確立したとします。クライアントで使用されている文字セットに関係なく、Adaptive Server 内で処理される文字はサーバのデフォルト文字セットに変換されます。Adaptive Server のデフォルト文字セットが “iso_1” だと仮定して、次のコマンドを考えてみます。

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

パスワードのパラメータは varchar であり、引用符で囲まれた文字列で表現され、暗号化される前に “iso_1” エンコーディングで保存されます。後で Adaptive Server のデフォルト文字セットが変更された場合でも、暗号化されたパスワードは元のデフォルト文字セットでエンコードされた文字列が暗号化されたもののままです。これでは文字のマッピングが一致しないので認証が失敗します。デフォルト文字セットの変更はめったにありませんが、プラットフォーム間での移行では重要な問題となります。

Adaptive Server はプラットフォーム、チップ・アーキテクチャ、文字セットなどの違いを超えてパスワードを使用できるように、クリア・テキストのパスワードを標準の形式に変換してから暗号化します。

パスワードが標準の形式に変換されてから syslogins に保存するには、次の手順に従います。

1 クリア・テキスト・パスワード文字列を UTF-16 に変換。

2 UTF-16 文字列をネットワーク・バイト順序に変換。

3 ランダム・バイトの小さなバッファをソルトとしてパスワードの末尾に付加。

4 SHA-256 ハッシュ・アルゴリズムを適用。

5 ダイジェスト、ソルト、およびバージョンを password カラムに保存。

認証過程は次のようになります。

1 クリア・テキスト・パスワード文字列を UTF-16 に変換。

2 UTF-16 文字列をネットワーク・バイト順序に変換。

3 syslogins の password カラムからのソルトをパスワードの末尾に付加。

4　ハッシュ・アルゴリズムを適用。

5　その結果を syslogins の password カラムと比較して、一致したら認証の成功。

アップグレードとダウングレードの動作

この項では、Adaptive Server のバージョン間におけるアップグレードとダウングレードについて説明します。

注意　Adaptive Server バージョン 15.0 からバージョン 15.7 以降にアップグレードする場合は、この項を確認してください。

ログイン・パスワードのダウングレード

15.0.2 より前のバージョンから新しいディスク・ベースの暗号化アルゴリズムへの移行を容易にするために、Adaptive Server には allow password downgrade (パスワードのダウングレードを許可) というパスワード・ポリシーが用意されています。15.0.2 より前のバージョンからアップグレードすると、このポリシーの値は “1” となり、パスワードが旧バージョンで使用された Sybase 独自のアルゴリズムと Adaptive Server 15.0.2 以降で使用される新しい SHA-256 アルゴリズムの両方で保管されることを示します。

パスワードが新旧両形式で保管されている限り、Adaptive Server を、ユーザ・パスワードをリセットせずに Adaptive Server 15.0 にダウングレードできます。allow password downgrade ポリシーが 0 に設定されると、パスワードは SHA-256 形式のみで保管され、旧バージョンとの互換性がなくなります。以前のバージョンにダウングレードするとき、SHA-256 で保管されているパスワードのみがランダム・パスワードにリセットされて旧バージョンと互換性のある古い形式で保管されます。

パスワードのダウングレードを許可する期間を終了するには、次のコマンドを実行します。

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

このコマンドを実行する前に、sp_displaylogin を使用してログイン・アカウントを調べ、使用されていたアカウントかどうかとパスワードが SHA-256 エンコーディングで保管されているかどうかを確認する必要があります。そうでない場合、そのアカウントは自動的にロックされ、パスワードは生成されたパスワードにリセットされます。そのアカウントを再度使用できるようにするには、アカウントをロック解除してユーザに新しく生成されたパスワードを通知する必要があります。

このコマンドの出力にはロックされたログイン・アカウントの情報とそのアカウント用に生成されたパスワードが含まれている場合があるので、保存しておく必要があります。

パスワードのダウングレード期間が終了すると、次の動作が行われます。

- master.dbo.sysattributes にパスワードのダウングレード期間が終了した datetime が記録されます。
- syslogins 内の各 password カラムの値は新しいパスワード・オンディスク構造のみを含むように書き換えられます。
- 新しいアルゴリズムへ移行していないログインは、SHA-256 フォーマットを使用してサーバが新しく生成したパスワードでリセットされ、ロックされます。生成されたパスワードは上記の sp_passwordpolicy プロシージャを実行している管理者にのみ表示されます。ロックの理由は 3 (「ログインまたは役割が SHA-256 に移行していなかった」) に設定されます。

sp_passwordpolicy プロシージャが完了した後は、次の動作が行われます。

- ログイン認証は SHA-256 のみを使用します。
- ディスク構造上の新しいパスワードのみが password カラムで使用されます。
- ロックされているログインを使用すると認証は失敗します。 ロックされたログインを使用するには、sp_locklogin を実行してそのログインをロック解除する必要があります。パスワードは sp_passwordpolicy によって生成されたものを使用します。ロックされたログイン・アカウントのパスワードには、生成されたものを使用せずに新たに割り当てることもできます。

例 1

この例ではアップグレードされたサーバを SHA-256 専用にする準備をします。ログイン・アカウントを調べて各アカウントでどの暗号化方法が使用されているかを確認するために sp_displaylogin を使用します。

```
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 の行に表示された値 SYB-PROP は、そのアカウントでは Sybase 独自の暗号化のみが使用されていることを示します。このログインは Adaptive Server バージョン 15.0.2 以降にアップグレードされる前に使用されておらず、sp_passwordpolicy 'set', 'allow password downgrade', '0' が実行されると、ロックされパスワードがリセットされます。

Adaptive Server 15.0.2 にアップグレードした後でログインがあったアカウントでは、新旧両暗号化が使用できることを示すように行が変更されます。

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

アクティブなログイン・アカウントがすべてこの状態になっているのが望ましい状態です。その場合、sp_passwordpolicy 'set', 'allow password downgrade', '0' を実行しても、ロックされてパスワードがリセットされるアカウントの心配をする必要はありません。

sp_passwordpolicy 'set', 'allow password downgrade', '0' を実行した後は、SHA-256 暗号化のみが使用されるようになり、次のような行が表示されます。

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

この値を示すログイン・アカウントは、ディスク・ベースの強力な暗号化アルゴリズムを使用するようになります。

すべてのパスワードが新しいアルゴリズムを使用するように変更された後では、sp_passwordpolicy を再実行してもリセットまたはロックされるアカウントはありません。

```
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)
```

例 2

この例では、1000 あるログイン・アカウントの中の 990 は SHA-256 アルゴリズムに移行していますが、残りの 10 アカウントはまだ SYB-PROP アルゴリズムを使用しています。

```
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)
```

注意 ログイン名、suid、および生成されたパスワードが、プロシージャを実行している管理者に表示されます。コマンドの出力として、リセットされてロックされた移行前の 10 アカウントすべてが表示されます。

アップグレードされた *master* データベースの動作の変化

master データベースをアップグレードする場合、Adaptive Server は password カラム内の Adaptive Server の以前のバージョンとアップグレード・バージョンのアルゴリズムを使用して、syslogins カタログ内の暗号化されたパスワードを維持します。

ユーザは sp_displaylogin を呼び出してどの “Login password encryption” がログインで使用されるかを調べることができます。

アップグレード後の最初のログイン認証では

- ユーザは認証に password カラムの内容と古いアルゴリズムを使用します。
- Adaptive Server は password カラムを古い暗号化アルゴリズムを使用して更新し、その後で新しい暗号化アルゴリズムを使用して更新します。

アップグレード後、次のログイン認証では、“allow password downgrade” が 0 に設定される前は、ユーザの認証に新しいアルゴリズムが使用されます。

新しい *master* データベースの動作の変化

新しい Adaptive Server master データベースでも、allow password downgrade を 0 に設定した後のアップグレード版 master データベースでも、password カラム内の新しいアルゴリズムのみを使用して暗号化されたパスワードが syslogins 内に維持されます。接続要求の認証とディスクへのパスワードの保管には SHA-256 アルゴリズムのみが使用されます。

サーバがアップグレードされ (バージョン 15.0 から 15.0.2 へのアップグレードなど)、アップグレード前とアップグレード後のサーバのアルゴリズムを使用してパスワードを維持しているかどうか、またはサーバが新しくインストールされ、(15.0.2 バージョンの) 最新のアルゴリズムを使用する master データベースが含まれているかどうかを特定するには、sp_passwordpolicy を発行します。

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

アップグレードしてからダウングレードした後のパスワード暗号化の保持

Adaptive Server 15.0.2 以降にアップグレードしてから、前のバージョンにダウングレードする場合は、sp_downgrade を使用して 15.0.2 以降のサーバのパスワード暗号化機能を保持します。デフォルトでは、Adaptive Server は、パスワードのダウングレード期間が終了するまで、アップグレード後にパスワードをダウングレードできます。

注意 sp_downgrade を実行しサーバをシャットダウンした後で、ダウングレードした同じバージョンの Adaptive Server を再起動すると、sp_downgrade で加えられた変更が削除されます。sp_downgrade を再実行して、その変更を再実行する必要があります。sp_downgrade の実行については、『インストール・ガイド』を参照してください。

アップグレード前の領域の追加

Adaptive Server では、master データベースとトランザクション・ログに追加の領域が必要です。master データベースとトランザクション・ログにさらに領域を追加するには、alter database を使用してください。

暗号化アルゴリズムとパスワード・ポリシー

- syslogins に必要な領域を約 30% 増加します。
- ローの長さの最大値を 1 ログイン・アカウントあたり 135 バイト増加します。
- ページあたりのロー数を、Adaptive Server バージョン 15.0.1 と 15.0.2 との間で 2K ページあたり約 16 ローから 12 ローに減らします。ダウングレード中に、allow password downgrade の値が 1 になっている期間があります。この場合は、新旧両方のパスワード暗号化アルゴリズムが使用されるので、2K ページあたり約 10 ローまで減少します。

  たとえば、Adaptive Server 15.0.1 のログイン・アカウントが 1,000 あり、そのデータが 59 ページに収まっているとすると、同じ数のログイン・アカウントで Adaptive Server 15.0.2 の新しい master データベースは約 19 ページ増加することになり、15.0.1 からアップグレードして allow password downgrade の値が 1 に設定されている場合は 33 ページ増加することになります。

トランザクション・ログには、更新された password カラム用に追加する領域が必要になります。最初のログインでは、1,000 ログインあたり約 829 2K ページが必要です。アップグレードとダウングレード中にユーザが行うパスワードの変更には、1,000 ログインあたり約 343 ページが必要です。十分なログ領域を確保するために、ユーザが Adaptive Server 15.0.2 以降に初めてログインする場合は、パスワードのアップグレードまたはダウングレードを実行する前にログイン 1 件あたり 1 ページ (約 2K ページ) の空きログ領域があることを確認してください。

ダウングレード

Adaptive Server では、バージョン 15.0.2 以降からバージョン 15.0 または 15.0.1 へのダウングレードをサポートしています。Adaptive Server の前のバージョンにダウングレードする場合は、追加作業が必要になる場合があります。

allow password downgrade が 0 または NULL になっている、または、パスワードが SHA-256 アルゴリズムのみで syslogins に保管されている場合は、sp_displaylogin をログイン・アカウントに適用して使用されているアルゴリズムを調べることができます。リセットされるアカウントを確認するには、sp_downgrade “prepare” を使用します。

prepare オプションでは、サーバをダウングレードする準備ができているかどうかが報告されます。prepare オプションが失敗すると、修正が必要なエラーが報告されます。エラーが修正される前にサーバでダウングレードが実行されると、ダウングレードは失敗します。ログイン・パスワードに関しては、prepare によって、どのパスワードがダウングレード中にリセットされるかが報告されます。

sp_downgrade を実行する必要があるかどうかを確認するには、sp_downgrade “prepare” を実行します。

```
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
```

データベースにそのログインのユーザ・エントリがある場合は、master データベースを使用してデータベースからそのユーザを削除し、その後でログインを削除します。

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

probe ログインはダウングレードされたサーバで installmaster を実行したときに再度作成されます。

sp_downgrade を実行する前に、Sybase では syslogins と syssrvroles の統計を削除するようおすすめしています。この操作を行うのは、パスワード・カラムの長さなど、sysstatistics 内の無効なカラム情報がダウングレード中に記録されるのを避けるためです。

syslogins と syssrvroles の統計を削除するには、次の行を入力します。

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

この例では、sp_downgrade を実行することで、user103 のログイン・パスワードがロックされリセットされています。Adaptive Server によって生成されたランダム・パスワードは sp_downgrade を実行しているクライアントにのみ表示されます。管理者はこの出力をファイルにリダイレクトして、そのパスワードを保存できます。ダウングレードを完了し、サーバを再起動した後に手動リセットすることもできます。

```
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)
```

追加のメッセージがエラー・ログに表示され、sp_downgrade の実行過程を確認できます。

```
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 はカタログの変更とパスワード・データの変更を行います。sp_downgrade の実行を成功させるには、サーバをシングル・ユーザ・モードにする必要があります。サーバをシングル・ユーザ・モードで再起動し、システム管理者だけがログインできるようにするには、-m コマンド・ライン・オプションを使用してサーバを起動します。

sp_downgrade を実行した後で、データやシステム・カタログを変更する可能性のある新しいログインやその他のアクションを避けるには、15.0.2 サーバをシャットダウンします。sp_downgrade を実行した後に、Adaptive Server をバージョン 15.0.2 で再起動すると、前のバージョンがシャットダウンし、バージョン 15.0.2 以降のレベルに再度アップグレードされます。

allow password downgrade を 0 に設定したときパスワードを無効にする方法

パスワード・ダウングレード期間の終了時に、syslogins のパスワードを有効期限切れにします。

ログイン・パスワードが無効になるように設定するには、次のコマンドを使用します。

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

役割パスワードが無効になるように設定するには、次のコマンドを使用します。

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

アクティブでないログイン・パスワードが無効になるように設定するには、次のコマンドを使用します。

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

アクティブでない役割パスワードが無効になるように設定するには、次のコマンドを使用します。

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

sp_passwordpolicy “expire stale login passwords” の *datetime* パラメータで設定した日付以降に変更されていないパスワードは、コマンドの実行時に期限切れになります。ユーザは、パスワード・ダウングレード期間の終了後にパスワードを自動的に変更する必要があります。

アクティブでないログインや役割をロックすることもできますが、正規のユーザがそのログイン・アカウントに再びアクセスできるようにするには、手動でパスワードをリセットする必要があります。

allow password downgrade の現在の設定値を表示する方法

allow password downgrade の現在の設定値を取得するには、次のように入力します。

```
sp_passwordpolicy ‘list’, ‘allow password downgrade’
```

結果セットには現在の値とその意味を説明するメッセージが含まれています。

master データベースをアップグレードし、パスワードを新旧両エンコーディングで維持している場合は、次のような結果が出力されます。

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

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

新しいパスワード暗号化のみを使用するアップグレードされた master データベースの場合、次のような結果が出力されます。

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

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

新しいパスワード暗号化のみを使用する Adaptive Server 15.0.2 の新しい master データベースの場合は、次のような結果が出力されます。

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

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

高可用性環境でのパスワードの使用

パスワード・セキュリティは高可用性の設定およびプライマリ・サーバとコンパニオン・サーバ間での syslogins におけるパスワードの動作に影響します。

高可用性の設定

高可用性を設定する前に、プライマリ・サーバとコンパニオン・サーバの allow password downgrade の値が同じになっている必要があります。allow password downgrade quorum 属性は、allow password downgrade の値がプライマリ・サーバとセカンダリ・サーバの両方で同じになっているかどうかをチェックします。

プライマリ・サーバでは allow password downgrade が 1 に、セカンダリ・サーバでは 0 に設定されていると、sp_companion の出力は、次のようになります。

```
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 カラムの値が 2 になっていますが、これは両サーバの値が一致していないので、ユーザがクラスタ・オペレーションを進めることができないことを示します。

sp_companion do_advisory も両サーバでの “allow password downgrade” の値の違いを表示します。

値を同期させ、両サーバが同じ状態であることを確認するには、sp_passwordpolicy ‘allow password downgrade’ をプライマリ・サーバとセカンダリ・サーバで別々に実行する必要があります。

アップグレード後に更新されたパスワード

高可用性を実現するためにアップグレードと設定を行った後で、プライマリ・サーバへの初回の接続が確立されると、ユーザ・ログインのパスワードは、同じオンディスク暗号化フォーマットを使用して、プライマリ・サーバとコンパニオン・サーバの両方で同期化されます。こうしておくと、allow password downgrade 期間が終了し、パスワードが以前の Adaptive Server バージョンにダウングレードされたときに、パスワードのリセットやロックを避けることができます。ログイン・パスワードは sp_passwordpolicy や sp_downgrade によるリセットやロックを避けて使用を継続できます。

高可用性環境のセットアップに成功した後、allow password downgrade 期間をプライマリ・サーバとコンパニオン・サーバで別々に終了します。以前のバージョンの Adaptive Server にダウングレードする必要があるときにも同様に sp_downgrade をプライマリ・サーバとコンパニオン・サーバで別々に実行します。

パスワードとログイン・ポリシーの設定

Flk Adaptive Server には、内部認証のログイン、役割、およびパスワードのポリシーを設定する制御がいくつか用意されています。

システム・セキュリティ担当者は、Adaptive Server で次の設定を行えます。

- 無効なパスワードが何回入力されたらログインや役割を自動的にロックするか、その回数を指定する。
- パスワードが失われた場合のログイン
- 手動によるログインと役割のロックとロック解除
- ログイン・パスワード情報の表示
- サーバワイドまたは特定のログインや役割に対する最小パスワード長 (minimum password length) の指定
- 複雑なログインのパスワードのチェック
- ログインのカスタムのパスワード・チェックの有効化
- パスワード有効期間の設定
- ログインのパスワード文字セットを考慮する。
- 非アクティブなログイン・アカウントのロック
- パスワードを高可用性環境で使用する。

失敗したログイン

ユーザが Adaptive Server のデータにアクセスするためには、Adaptive Server によって認証される必要があります。認証が失敗した場合は、Adaptive Server から次のメッセージが返され、ネットワーク接続が終了します。

```
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
```

このメッセージはログインの失敗を示す汎用のメッセージであり、接続中のユーザに対して、ユーザ名やパスワードの誤りが原因でログインが失敗したかどうかは通知しません。

クライアントには、悪意のあるユーザに情報を提供しないように、ログインの失敗を示す汎用のメッセージが表示されますが、システム管理者にとっては、失敗の理由が侵入の試行の検出やユーザ認証の問題の診断に役立つ重要なものである場合があります。

Adaptive Server は、sysaudits.extrainfo カラムの `Other Information` の項にある `Errornumber.Severity.State` にログインの失敗の理由を表示します。ログイン失敗の監査には、イベント番号 45 と eventmod 2 が含まれています。

ログイン失敗の監査を有効にするには、sp_audit の login パラメータを on または fail に設定します。

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

「ログイン失敗の監査」を参照してください。

Adaptive Server ログイン・アカウントおよび役割のロック

ユーザが Adaptive Server にログインできないようにするには、Adaptive Server ログイン・アカウントをロックするか、削除します。ログイン・アカウントをロックすると、suid は維持され、再利用はできません。

ログイン・アカウントをロックするには、sp_locklogin を実行します。

ログイン試行回数が maximum failed login の設定値に到達したためにログイン・アカウントがロックされると、AUD_EVT_LOGIN_LOCKED (112) 監査イベントのある監査レコードが login_locked 監査オプションによって生成されます。

警告！ 削除されたログイン・アカウントのサーバ・ユーザ ID (suid) は、次にログイン・アカウントが作成されるときに再利用される場合があります。このことが発生するのは、削除されるログインの suid が、syslogins 内で最大である場合だけです。しかし、drop login の実行が監査されない場合には、このことによって責任に関する問題が発生する可能性があります。また、再利用された suid を持つユーザが、その古い suid に認可されていたデータベース・オブジェクトにアクセスできるようになるというおそれもあります。

次の場合は、ログインは削除できません。

- ユーザがいずれかのデータベースを使用している場合。
- そのログインが、システム・セキュリティ担当者またはシステム管理者の役割を保持している最後に残ったユーザである場合。

システム・セキュリティ担当者は、sp_locklogin または drop login を使用して、ログインをロックまたは削除することができます。システム・プロシージャが複写用にログに記録されている場合、システム・セキュリティ担当者は、コマンドの発行時に master データベース内になければなりません。

ログインのロックとロック解除

次のような場合にログインをロックできます。

- パスワードの期限が切れた。
- ログインを試行できる最大回数に達した。
- システム・セキュリティ担当者が手動でロックした。

❖ ログインのロックとロック解除

- システム・セキュリティ担当者は、sp_locklogin を使用して、ログインを手動でロックまたはロック解除することができます。次に例を示します。

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

  ログインのロック・ステータスに関する情報は、syslogins の status カラムに格納されます。

  『リファレンス・マニュアル：プロシージャ』の「sp_locklogin」を参照してください。

ログイン・アカウントのロックとロック解除

sp_locklogin を使用すると、アカウントのロックとロック解除、ロックされているアカウントのリストの表示ができます。sp_locklogin を使用できるのは、システム・セキュリティ担当者だけです。

構文は次のとおりです。

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

各要素の意味は次のとおりです。

- *login_name* には、ロックまたはロック解除するアカウントの名前を指定します。ログイン名は既存の有効なアカウントでなければなりません。
- all は、sa_role を除く、Adaptive Server の全ログイン・アカウントのロックまたはロック解除を指示します。

- lock | unlock はアカウントのロックまたはロック解除を指定します。

ロックされているすべてのログインの一覧を表示するには、パラメータを指定しないで sp_locklogin を実行します。

既にログインしているアカウントをロックすることもできますが、そのユーザがアカウントを使用できなくなるのはログアウトした後です。データベース所有者のアカウントをロックし、ロックされたアカウントがデータベース内のオブジェクトを所有するようにすることができます。sp_changedbowner を使用して、ロックされているアカウントをデータベースの所有者として指定できます。

Adaptive Server では、ロックされていないシステム・セキュリティ担当者アカウントとロックされていないシステム管理者アカウントが少なくとも 1 つずつ常に存在することが保証されます。

アカウントがロックされているかどうかを追跡する場合の *syslogins* の使用

syslogins には、lastlogindate、crdate、locksuid、lockreason、および lockdate カラムが含まれており、最後のログインと非アクティブ・アカウントのロックをサポートします。アカウントの所有者または管理者は、アカウントがロックされているか、いつロックされたか、誰がロックしたか、なぜロックされたかを知ることができます。

ログイン作成時に、crdate カラムはそのときの日時に設定されます。

enable last login updates パスワード・ポリシー・オプションが 1 に設定されている場合、lastlogindate カラムはログインの datetime に設定され、そのカラムの以前の値がそのログイン・セッションのプロセス・ステータス構造体に保存されます。syslogins とプロセス・ステータス構造体の更新は Adaptive Server にログインするたびに行われます。新しい master データベースまたはアップグレードされたデータベースでの enable last login updates のデフォルト値は 1 です。このオプションを無効にするには、管理者の権限を使用してプロシージャを実行します。

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

@@lastlogindate は各ユーザ・ログイン・セッションに固有のもので、そのアカウントへの前回のログイン日時を知るために各セッションで使用できます。以前に使用されたことのないアカウントの場合、または enable last login updates が 0 に設定されている場合、*@@lastlogindate* の値は NULL です。

トランザクション・ログは、syslogins..lastlogindate の更新のログを取りません。

sso_role パーミッションが付与された管理者は、次のようにして、所定日数の間、非アクティブになっているログイン・アカウントをロックできます。

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

このコマンドは、enable last login updates が 0 に設定されている場合、あるいは lastlogindate カラムの値が NULL になっている場合は、何もしません。*number of inactive days* の値の範囲は、1 ～ 32767 (日) です。

lockreason カラムは、ログインがロックされた理由を指定します。lockdate カラムの値はそのときの datetime に設定されます。

ロック解除されたアカウントの lockreason、lockdate、locksuid の各カラムは NULL にリセットされます。

lockdate、locksuid、lockreason の各カラムの設定は Adaptive Server 内部で処理されます。表 3-3 は lockreason の値と説明、さらに locksuid の値を示します。

表 3-3: locksuid の値と理由

lockreason の値	locksuid の値	lockreason アカウントの説明
NULL	NULL	アカウントはロックされていない。
0	sp_locklogin の呼び出し元の suid	locksuid が sp_locklogin を手動で実行してアカウントをロックした。
1	sp_locklogin の呼び出し元の suid	アカウントは非アクティブだったため locksuid が sp_locklogin 'all', 'lock', 'ndays' を手動で実行してロックした。
2	ログイン試行の suid	アカウントは、失敗ログイン数が許容最大数に達したため Adaptive Server によってロックされた。
3	sp_passwordpolicy set, "allow password downgrade", 0 の呼び出し元の suid	アカウントは、ログインまたは役割がパスワード・ダウングレード期間の終了にもかかわらず SHA-256 に移行していなかったので、locksuid によってロックされた。
4	NULL	アカウントが非アクティブであったためにロックされた。

役割のロックとロック解除

役割がロックされた日時、ロックされた理由、ロック実施者などのアカウント情報は syssrvroles に保存されているので、役割がロックされているアカウントに用いると役立ちます。

次のように、役割がロックされるのにはいくつかの理由があります。

- 誤った役割パスワードを指定されている回数分入力した。役割の作成時または変更時、役割に 'max failed_logins' オプションを関連付けることができます。このオプションでは、役割がロックされるまでの役割アクティブ化試行失敗回数が指定されます。
- 次のように、alter role を使用して役割を手動でロックした。

```
alter role rolename lock
```

Adaptive Server には syssrvroles 内にロック情報用のカラムがあります。次にそのカラムを示します。

- lockdate － 役割がロックされた日付を示します。
- locksuid － 役割のロックを実行した人物を示します。
- lockreason － 役割がロックされた理由を示します。次に示すように、これはコード形式です。

lockreason の値	locksuid の値	役割の lockreason の説明
NULL	NULL	役割はロックされていません。
1	alter role の呼び出し元の suid	alter role rolename lock を手動で実行することにより suid によって役割がロックされます。
2	役割がロックされる原因となった役割のアクティブ化を最後に試行したユーザの suid	役割アクティブ化の試行回数が max failed logins に達したため Adaptive Server によって役割がロックされました。

❖ **役割のロックとロック解除**

- システム・セキュリティ担当者は、alter role を使用して、役割を手動でロックまたはロック解除することができます。次に例を示します。

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

役割のロック・ステータスについての情報は、syssrvroles の status カラムに格納されます。

『リファレンス・マニュアル：コマンド』の「alter role」を参照してください。

注意 可用性の高い環境では、これらの syssrvrole カラムはプライマリ・サーバとセカンダリ・サーバの両方で更新されます。

スレッショルドを所有するログインのロック

この項では、スレッショルドについて説明し、ロックされたユーザ・ログインからスレッショルドが受ける影響について説明します。

- スレッショルド・ストアド・プロシージャは、セキュリティの手段として、そのプロシージャを作成したログインのアカウント名と役割を使用して実行されます。
  - スレッショルドを所有するユーザのログインは削除できません。
  - スレッショルドを所有するユーザのログインをロックすると、ユーザはストアド・プロシージャを実行できません。
- ラストチャンス・スレッショルドと “sa” ログインが作成したスレッショルドは、sp_locklogin の影響を受けません。“sa” ログインをロックしても、ラスト・チャンス・スレッショルドと “sa” ユーザが作成または修正したスレッショルドは起動します。

ログイン・プロファイルの管理

ログイン・プロファイルの定義、変更、および削除は、システム・セキュリティ担当者が実行できます。

表 3-4 には、ログイン・プロファイルの作成および管理に使用するシステム・プロシージャがまとめられています。

表 3-4: Adaptive Server でのログイン・プロファイルの管理

タスク	必要な役割	コマンドまたはプロシージャ	データベース
ログイン・プロファイルの作成	システム・セキュリティ担当者	create login profile	マスタ・データベース
ログイン・プロファイルの変更	システム・セキュリティ担当者	alter login profile	マスタ・データベース
ログイン・プロファイルの削除	システム・セキュリティ担当者	drop login profile	マスタ・データベース
ログイン・プロファイル ID の出力	システム・セキュリティ担当者	lprofile_id	任意のデータベース
ログイン・プロファイル名の出力	システム・セキュリティ担当者	lprofile_name	任意のデータベース
ログイン・プロファイル名の表示	システム・セキュリティ担当者	sp_displaylogin	任意のデータベース
ログイン・プロファイル情報の表示	システム・セキュリティ担当者	sp_securityprofile	任意のデータベース

ログイン・プロファイルの属性

表 3-5 には、ログイン・プロファイルの属性がまとめられています。ログイン・プロファイルの属性は、syslogins、sysloginroles、および master.dbo.sysattributes に格納されます。

表 3-5: ログイン・プロファイルの属性

属性	説明
default database	Adaptive Server のデフォルト・データベース。
default language	デフォルトの言語。
login script	有効なストアド・プロシージャ。create login、alter login、create login profile、および alter login profile によってログイン・スクリプトとして使用されるストアド・プロシージャは、120 文字に制限されています。
auto activated roles	ログイン時に自動的にアクティブ化する必要のある、パスワード保護されていない、以前に付与されているユーザ定義の役割。指定された役割がログインを許可されていない場合、エラーが生成されます。デフォルトでは、ユーザ定義の役割はログイン時に自動的にはアクティブ化されません。
authenticate with	ログイン・アカウントの認証に使用するメカニズムを指定します。 authenticate with *authentication mechanism* を指定しない場合、ログイン・アカウントには ANY の値が使用されます。
track lastlogin	最終ログイン更新を有効にします。
stale period	ログイン・アカウントがロックされるまでに非アクティブ状態であることができる期間を示します。
profile id	Adaptive Server のデータベースを指定します。

ログイン・プロファイルとパスワード・ポリシー属性の適用

大量のログイン・アカウントの属性は、1 つのログイン・プロファイルをすべてのログイン・アカウントのデフォルトとして、ログイン・アカウントのサブセットとして、または個別のログイン・アカウントとして定義することによって管理できます。

ログイン・プロファイルの属性は、次の優先度でログイン・アカウントに関連付けられます。

1 ログインにバインドされているログイン・プロファイルの属性値

2 デフォルトのログイン・プロファイルの属性値

3 次の状況で sp_passwordpolicy を使用して指定されている値

- デフォルトのログイン・プロファイルが存在していない。
- ログイン・プロファイルが定義されていない。アカウントへのバインドもされていない。
- ログイン・プロファイルが無視されるように設定されている（create login コマンドに with login profile ignore パラメータが指定されている)。

4 属性のデフォルト値

ログイン・プロファイルの作成

次の手順では、特定のサーバでのログイン・プロファイルおよびログイン・アカウントの作成と、ユーザ・パーミッションの管理について説明します。

1 ログイン・アカウントのログイン・プロファイルを作成するのはシステム・セキュリティ担当者です。

2 システム・セキュリティ担当者は新しいユーザのログイン・アカウントを作成し、そのログイン・プロファイルを新しいログイン・アカウントに関連付けます。

3 システム管理者またはデータベース所有者が、ユーザをデータベースに追加するかグループに割り当てます。

4 ユーザまたはログイン・プロファイルに特定の役割を付与するのもシステム・セキュリティ担当者です。

5 システム管理者、データベース所有者、またはオブジェクト所有者が、特定のコマンドとデータベース・オブジェクトに対するパーミッションを、ユーザまたはグループに付与します。

次の例では、ログイン・プロファイル mgr_lp を作成します。

```
create login profile mgr_lp
```

『リファレンス・マニュアル：コマンド』の「create login profile」を参照してください。

デフォルトのログイン・プロファイルの作成

次の例では、emp_lp という名前のデフォルトのログイン・プロファイルを作成します。現在、別のログイン・プロファイルがデフォルトのログイン・プロファイルとして設定されている場合、そのデフォルト・プロパティは解除されて emp_lp に適用されます。

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

『リファレンス・マニュアル：コマンド』の「create login profile」を参照してください。

ログイン・プロファイルのログイン・アカウントとの関連付け

ログイン・アカウントの作成時にログイン・プロファイルが指定されない場合、デフォルトのログイン・プロファイルがその新しいアカウントに関連付けられます。デフォルトのログインが存在しない場合は、sp_passwordpolicy によって指定されるパスワード・ポリシー属性かデフォルトの属性が適用されます。属性が適用される順序の詳細については、「ログイン・プロファイルとパスワード・ポリシー属性の適用」を参照してください。

次の例では、パスワードを rubaiyat としてログイン・アカウント omar_khayyam を作成し、このアカウントにログイン・プロファイル emp_lp を関連付けます。

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

次の例では、ログイン・アカウント omar_khayyam を変更し、このアカウントにログイン・プロファイル staff_lp を関連付けます。

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

ログイン・プロファイルの無視

ignore login profile 句は、関連付けられているデフォルトのログイン・プロファイルを直接またはデフォルトのログイン・プロファイルから無効にするときに使用します。Adaptive Server は、ログイン・アカウントの対応する属性を適用するための優先度の規則に従います。詳細については、「ログイン・プロファイルとパスワード・ポリシー属性の適用」を参照してください。

次の例では、ログイン・アカウントを作成して任意のログイン・プロファイルを無視するように指定します。

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

新しいログイン・プロファイルへの既存のログイン・アカウント値の転送

次の例では、既存のログイン・アカウントの値を新しいログイン・プロファイルに転送する方法を示します。ログイン・プロファイル sa_lp が、ログイン・アカウント ravi と同じ値に設定された default database、default language、authenticate with の各属性値を使用して作成されます。

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

ログイン・プロファイルの手動での複写

プロファイル ID は新しいログイン・プロファイルの ID を指定する属性で、Adaptive Server では、ログイン・プロファイルを手動で複写する際に使用されます。

たとえば、プロファイル ID に 25 が指定されているプロファイル emp_lp をレプリケート master で作成する場合は、次のコマンドを実行します。

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

ログイン・プロファイルへの役割の付与

次の例では、ログイン・プロファイル def_lp を作成し、このログイン・プロファイルに役割 access_role を付与します。

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

def_lp にバインドされているログインは、暗黙的に access_role が付与されます。システム・セキュリティ担当者は、ログイン・プロファイルに付与された役割をバインドされているログインのデフォルトの役割として動作するように指定することができます。つまり、ログイン時のユーザ・セッションでその役割が自動的にアクティブ化されます。

自動的にアクティブ化される役割の追加または削除の詳細については、「自動的にアクティブ化された役割の追加または削除」(63 ページ) を参照してください。

ログイン・スクリプトの起動

ログイン・スクリプトを、ログイン時にログイン・プロファイルから起動するように指定することができます。グローバル・ログイン・トリガが sp_logintrigger によって指定されている場合、そのログイン・スクリプトはグローバル・ログイン・トリガの後に起動されます。

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

- ログイン・スクリプトが存在するデータベースと所有者の名前を指定することによって、ログイン・スクリプトを修飾できます。データベースの名前を使って修飾しない場合は、master データベースよりもデフォルトのデータベースが優先されます。
- 指定のログイン・スクリプトが所有者名で修飾されない場合は、ログイン・トリガが存在するデータベースの所有者よりも現在のログインであるログイン・トリガの所有者が優先されます。
- create login、alter login、create login profile、および alter login profile によってログイン・スクリプトとして使用されるストアド・プロシージャは、120 文字に制限されています。

詳細については、「ログイン・トリガの使用」(226 ページ) を参照してください。

ログイン・プロファイル情報の表示

この項では、ログイン・プロファイルに関する情報の表示方法について説明します。

ログイン・プロファイル名の表示

指定のログイン・プロファイル ID またはログイン suid を表示するには、次の構文を使用します。

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

指定のログイン ID が現在のユーザのログイン ID ではない場合、システム・セキュリティ担当者は指定のログイン ID のプロファイル名を表示する必要があります。

次は、指定のログイン・プロファイル ID のログイン・プロファイル名を表示します。

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

パラメータが指定されていない場合、現在のユーザのログイン・プロファイル名が返されます。ログイン・プロファイルが指定のログイン・アカウントに関連付けられていない場合は、デフォルトのログイン・プロファイルのログイン・プロファイル名が返されます。login profile ignore パラメータを設定する必要はありません。

ログイン・プロファイル名は、sp_displaylogin を使用して表示することもできます。ログイン・プロファイルがログイン・アカウントに直接関連付けられおらず、デフォルトのログイン・プロファイルが存在する場合は、デフォルトのログイン・プロファイルの名前が表示されます。

ログイン・プロファイル ID の表示

指定のログイン・プロファイル名またはログイン名のログイン・プロファイル ID を表示するには、次の構文を使用します。

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

指定のログイン名が現在のユーザのログイン名ではない場合、システム・セキュリティ担当者は指定のログイン名のプロファイル ID を表示する必要があります。

次は、指定のログイン・プロファイル ID のログイン・プロファイル名を表示します。

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

ログイン・プロファイルが指定のログイン・アカウントに関連付けられていない場合は、デフォルトのログイン・プロファイルのプロファイル ID が返されます。login profile ignore パラメータを設定する必要はありません。

ログイン・プロファイルのバインド情報の表示

ログイン・アカウントに関連付けられているログイン・プロファイルの属性を表示するには、sp_securityprofile を使用します。

注意 非特権ログイン・アカウントが表示できるのは、直接関連付けられているログイン・プロファイルの属性か、デフォルトのログイン・プロファイルの属性のみです。システム・セキュリティ担当者は、すべてのログイン・プロファイルの属性とバインドを確認する必要があります。

構文の詳細については、『リファレンス・マニュアル：システム・プロシージャ』の「sp_securityprofile」を参照してください。

ログイン・プロファイルの修正

ログイン・プロファイルの属性とその属性に対応する値を追加、削除、または変更するには、alter login profile コマンドを使用します。属性が指定されていない場合は、ログイン・プロファイルに追加されます。ログイン・プロファイルの属性のリストについては、「ログイン・プロファイルの属性」(57 ページ)を参照してください。

次の例では、ログイン・プロファイル mgr_lp からログイン・スクリプト属性を削除します。ログイン・スクリプトがデフォルトのログイン・プロファイルに指定されている場合、ログイン・スクリプトはログイン時に起動されます。それ以外の場合、ログイン・スクリプトは起動されません。

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

完全な構文については、『リファレンス・マニュアル：コマンド』の「alter login profile」を参照してください。

自動的にアクティブ化された役割の追加または削除

パスワード保護されていない、以前に付与されているユーザ定義の役割は、ログイン時に自動的にアクティブ化することができます。

次のコマンドでは、mgr_lp に関連付けられているユーザのログイン時に、ログイン・プロファイル mgr_lp を修正し、役割 mgr_role と eng_role を自動的にアクティブ化します。

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

ログイン・プロファイルに付与されるユーザ定義の役割のうち、自動的にアクティブ化される役割のステータスは、sysloginroles.status カラムに示されます。値「1」は、付与された役割がログイン時に自動的にアクティブ化される必要があることを示します。役割を取り消すと sysloginroles 内でその役割に対応するローが削除され、役割がログイン時に自動的にアクティブ化されなくなります。Adaptive Server は、次のようにユーザのログイン・プロファイルに付与された役割を自動的にアクティブ化します。

1　デフォルトのログイン・プロファイルがアカウントに関連付けられている場合、デフォルトのログイン・プロファイルで指定されている自動アクティブ化役割が適用されます。

2　アカウントに直接関連付けられているログイン・プロファイルとデフォルトのログイン・プロファイルの両方が存在する場合は、アカウントに直接関連付けられているログイン・プロファイルで指定されている自動アクティブ化役割のみが適用されます。

ログイン・プロファイルのデフォルトのログイン・プロファイルへの変更

as [not] default 句は、ログイン・プロファイルをデフォルトのログイン・プロファイルとして割り当てたり削除したりするときに使用します。

次の文は、emp_lp という名前のログイン・プロファイルをデフォルトのログイン・プロファイルとして変更します。

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

次の文は、emp_lp という名前のログイン・プロファイルをデフォルトのログイン・プロファイルとして削除します。

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

ログイン・プロファイルの削除

drop login profile コマンドは、ログイン・アカウントにバインドされていないログイン・プロファイルを削除します。ログイン・アカウントにバインドされているログイン・プロファイルを強制的に削除するには、drop login profile with override を使用します。削除するログイン・プロファイルがログイン・アカウントにバインドされている場合、このログイン・アカウントはデフォルトのログイン・アカウントにバインドされます (デフォルトのログイン・アカウントが存在する場合)。login profile ignore 句が指定されているとこの句は削除され、デフォルトのログイン・プロファイルが存在していればログイン・アカウントに関連付けられます。

次の例では、ログイン・プロファイル eng_lp を、1 つまたは複数のログイン・アカウントにバインドされている場合でも強制的に削除します。

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

データベースへのユーザの追加

データベース所有者またはシステム管理者は、sp_adduser を使用して、特定のデータベースにユーザを追加できます。このユーザは、Adaptive Server ログインを既に持っていなければなりません。構文は次のとおりです。

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

上記のパラメータの意味は、次のとおりです。

- *loginame* には、既存のユーザのログイン名を指定します。
- *name_in_db* には、このユーザをデータベース内でログイン名とは異なる名前で認識する場合に、その名前を指定します。

  name_in_db を使用すると、ユーザ各自の設定に対応することができます。たとえば、Mary という名前の Adaptive Server ユーザが 5 人いる場合、その 5 人はそれぞれ異なるログイン名を持つ必要があります。たとえば、Mary Doe は “maryd” としてログインし、Mary Jones は “maryj” としてログインします。ただし、これらのユーザが同じデータベースを使用するのでなければ、各ユーザが個々のデータベース内では “mary” として認識されるようにすることもできます。

  name_in_db パラメータを指定しない場合、データベース内での名前は loginame と同じものになります。

注意 この機能は、「データベース内でのエイリアスの使用」(70 ページ) で説明するエイリアス機能とは異なります。エイリアスは、1 人のユーザの識別子とパーミッションを別の名前に対応付けるためのものです。

- *grpname* は、データベース内の既存のグループの名前です。グループ名を指定しない場合、そのユーザはデフォルト・グループ “public” のメンバになります。ユーザは、別のグループのメンバになっても、“public” グループのメンバであることに変わりはありません。詳細については、「ユーザのグループ・メンバシップの変更」(69 ページ) を参照してください。

sp_adduser システム・プロシージャは、現在のデータベース内の sysusers システム・テーブルに 1 つのローを追加します。ユーザのエントリがデータベースの sysusers テーブルにあれば、そのユーザは次のことができます。

- use *database_name* コマンドを発行して、そのデータベースにアクセスする。
- create login でデフォルト・データベースが指定された場合は、デフォルトではそのデータベースを使用する。
- alter login を使用して、そのデータベースをデフォルトにする。

次の例は、データベース所有者が、作成済みの技術グループ “eng” の “maryh” に対してアクセス・パーミッションを付与する方法を示しています。

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

次の例は、“maryd” にデータベースへのアクセス権を与え、このデータベースでの名前をログイン名と同じものにする方法を示しています。

```
sp_adduser maryd
```

次の例は、既存の “eng” グループに “maryj” を追加する方法を示しています。このとき、データベース内での名前をログイン名と同じにするために、新しいユーザ名の代わりに null を指定します。

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

データベースへのアクセス権を持つユーザであっても、データベース内でのデータの読み込み、データの変更、特定のコマンドの使用といった操作を実行するには、パーミッションが必要です。このようなパーミッションを付与するには、grant コマンドと revoke コマンドを使用します。これらのコマンドについては、「第 6 章 ユーザ・パーミッションの管理」 を参照してください。

“guest” ユーザのデータベースへの追加

データベースに “guest” というユーザを作成すると、Adaptive Server アカウントを持つすべてのユーザが「**guest**」ユーザとしてそのデータベースにアクセスできるようになります。データベース・ユーザまたはエイリアス・ユーザとして追加されていないユーザが、use *database_name* コマンドを発行すると、Adaptive Server は guest ユーザがあるかどうかを検索します。guest ユーザがある場合は、ユーザは guest ユーザのパーミッションが与えられ、データベースへのアクセスを許可されます。

データベース所有者は、sp_adduser を使用して、データベースの sysusers テーブルに guest エントリを追加できます。

```
sp_adduser guest
```

guest ユーザを削除するには sp_dropuser を使用します。詳細については、「ユーザの削除」(80 ページ) を参照してください。

master データベースから guest ユーザを削除すると、どのデータベースにもまだ追加されていないサーバ・ユーザは Adaptive Server にログインできなくなります。

注意 1 つのデータベースで複数のユーザが guest ユーザになることができますが、このときも、Adaptive Server はサーバ内でユニークな、ユーザのサーバ・ユーザ ID を使用して、各ユーザの実行記録を監査できます。「第 8 章 監査」を参照してください。

“guest” ユーザのパーミッション

“guest” は “public” の権限を継承します。データベース所有者とデータベース・オブジェクトの所有者は、grant と revoke を使用して、“guest” の権限を “public” の権限よりも拡大あるいは縮小することができます。「第 6 章 ユーザ・パーミッションの管理」を参照してください。

Adaptive Server をインストールすると、master..sysusers に guest エントリが作成されます。

ユーザ・データベースの “guest” ユーザ

ユーザ・データベースでは、データベース所有者が guest ユーザを追加することによって、すべての Adaptive Server ユーザにそのデータベースの使用を許可できます。このようにすれば、sp_adduser を使用して個々のユーザを明示的にデータベース・ユーザとして指定する必要はありません。

guest を使用する方法を使うと、データベースへのアクセスを許可する一方でデータベース・オブジェクトへのアクセスを制限できます。

たとえば、titles テーブルの所有者は、次のコマンドを実行することによって、“guest” 以外のすべてのデータベース・ユーザに titles テーブルに対する select パーミッションを付与できます。

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

インストールされているシステム・データベースの “guest” ユーザ

Adaptive Server は、guest ユーザを使用して、システム tempdb データベースとユーザが作成したテンポラリ・データベースを作成します。tempdb で作成されたテンポラリ・オブジェクトとその他のオブジェクトは、“guest” ユーザによって自動的に所有されます。sybsystemprocs、sybsystemdb、および sybsyntax データベースには “guest” ユーザが自動的に含まれます。

pubs2 と *pubs3* の “guest” ユーザ

サンプル・データベースの “guest” ユーザ・エントリを使用すると、Adaptive Server の新規ユーザは『Transact-SQL ユーザーズ・ガイド』の例を使用できます。サンプル・データベース内の guest には、次のような広範囲の権限が与えられます。

- すべてのユーザ・テーブルに対する select パーミッションとデータ変更パーミッション
- すべてのプロシージャに対する execute パーミッション
- create table、create view、create rule、create default、create procedure の各パーミッション

guest ユーザのサーバへの追加

システム・セキュリティ担当者は、create login を使用して、一時的に使用するユーザ (たとえば visitor) が使用するログイン名とパスワードを追加できます。通常、こうしたユーザには制限されたパーミッションを付与します。デフォルト・データベースを割り当てることもあります。

警告！ ビジタ・ユーザ・アカウントは、“guest” ユーザ・アカウントと同じものではありません。ビジタ・アカウントのユーザはすべて、同じサーバ・ユーザ ID を持ちます。したがって、個々のアクティビティを監査することはできません。これに対して、“guest” ユーザはそれぞれユニークなサーバ ID を持つため、個々のアクティビティの監査が可能となり、個々の責任が明確になります。複数のユーザがビジタ・アカウントを使用するように設定すると、個々の責任が不明確になるため、Sybase ではこれを行わないことをおすすめします。

---

create login を使用して、master..syslogins に “guest” という名前のビジタ・ユーザ・アカウントを追加することができます。この “guest” ユーザ・アカウントは、システムの “guest” ユーザ・アカウントよりも優先されます。sp_adduser を使用して “guest” という名前のビジタ・ユーザを追加すると、システムの “guest” ユーザを処理するように設計された sybsystemprocs や sybsystemdb などのシステム・データベースが影響を受けます。

リモート・ユーザの追加

リモート・アクセスを有効にすると、サーバ上のストアド・プロシージャを、別の Adaptive Server 上のユーザが実行できるようになります。リモート・サーバのシステム管理者と協力することによって、自分のサーバ上のユーザに対してリモート・サーバへの「リモート・プロシージャ・コール」の実行を許可することもできます。

リモート・プロシージャ・コールを使用できるようにするには、ローカル・サーバとリモート・サーバの両方を設定する必要があります。『システム管理ガイド 第 1 巻』の「第 7 章 リモート サーバの管理」を参照してください。

グループの作成

グループを利用すると、単一の文で複数のユーザにパーミッションを付与したり、取り消したりすることができます。また、ユーザの集まりに名前を付けることもできます。グループは、Adaptive Server のユーザが多い場合に特に役立ちます。

グループを作成してからデータベースにユーザを追加します。これは、sp_adduser はユーザをデータベースに追加するだけでなく、ユーザをグループに割り当てることもできるためです。

sp_addgroup を使用してグループを作成するには、システム管理者またはシステム・セキュリティ担当者の役割が必要か、データベース所有者である必要があります。構文は次のとおりです。

```
sp_addgroup grpname
```

必須パラメータであるグループ名は、識別子の規則に従って指定してください。システム管理者、システム・セキュリティ担当者、またはデータベース所有者は、sp_changegroup を使用して、グループへのユーザの割り当てと再割り当てができます。

たとえば、Senior Engineering グループを設定するには、グループの追加先のデータベースを使用しているときに、次のコマンドを実行します。

```
sp_addgroup senioreng
```

この sp_addgroup システム・プロシージャは、現在のデータベース内の sysusers に 1 つのローを追加します。したがって、データベース内の各グループは、各ユーザと同様、sysusers に 1 つのエントリを持つことになります。

ユーザのグループ・メンバシップの変更

システム管理者、システム・セキュリティ担当者、またはデータベース所有者は、sp_changegroup を使用してユーザの所属グループを変更できます。各ユーザは、すべてのユーザが常にそのメンバとなる “public” グループの他に、ただ 1 つのグループのメンバになることができます。

sp_changegroup を実行するには、次の条件を満たしていることが必要です。

- グループが既に存在している。
- ユーザが現在のデータベースに対するアクセス権を持っている (sysusers に登録されている)。

sp_changegroup の構文は次のとおりです。

```
sp_changegroup grpname, username
```

たとえば、ユーザ “jim” を現在のグループからグループ “management” に変更するには、次のコマンドを使用します。

```
sp_changegroup management, jim
```

ユーザを他のグループに割り当てることなく現在のグループから削除するには、次のように所属グループを “public” に変更します。

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

“public” という名前は予約語なので、引用符で囲んでください。このコマンドを実行すると、Jim の所属グループは “public” だけになります。

あるグループから別のグループに変更されたユーザは、元のグループに属していたときに持っていたすべてのパーミッションを失いますが、新しいグループに与えられているパーミッションを取得します。

ユーザの所属グループの割り当てはいつでも変更できます。

グループの設定とユーザの追加

システム・セキュリティ担当者、システム管理者、またはデータベース管理者は、sp_addgroup *group_name* を使用してグループを作成します。

グループ・レベルでは、パーミッションを付与および取り消すことができます。グループのパーミッションは、グループのメンバに自動的に渡されます。各データベースには、作成時にすべてのユーザが自動的に属する “public” という名前のグループが設定されています。sp_adduser を使用してユーザをグループに追加し、sp_changegroup を使用してユーザのグループを変更します。「ユーザのグループ・メンバシップの変更」(69 ページ) を参照してください。

グループには、sysusers テーブルに対応するエントリが存在する名前を指定します。データベースではグループとユーザを作成するのに同じ名前を使用することはできません (たとえば、“shirley” という名前のグループとユーザの両方を作成することはできません)。

データベース内でのエイリアスの使用

エイリアスを使うと、1 つのデータベース内で複数のユーザを同じユーザとして扱い、同じ権限を持たせることができます。この方法は、複数のユーザがデータベース所有者の役割を持てるようにする場合によく使用されます。データベース所有者は、setuser コマンドを使用することにより、そのデータベース内で別のユーザになり代わって作業できます。エイリアスは、ユーザの集合に 1 つの ID を与えるために使用することもできます。

たとえば、ある会社で複数の副社長が同じ権限と所有権で 1 つのデータベースを使用できるようにする必要があるとします。Adaptive Server とデータベースにログイン名 “vp” を追加して、副社長全員が “vp” としてログインするようにした場合は、それぞれのユーザを区別する方法はありません。そこで、それぞれが別の Adaptive Server アカウントを持つようにして、副社長全員のエイリアスをデータベース・ユーザ名 “vp” とします。

注意 1 つのデータベース内で複数のユーザが同じエイリアスを使用できますが、その場合も、各ユーザが実行するデータベース操作を監査することによって、個々の責任を明確にすることが可能です。「第 8 章 監査」を参照してください。

エイリアスを使用して得られる集合ユーザ ID は、データベース・オブジェクトの集合所有権を意味します。たとえば、ユーザ “loginA” がデータベース db1 の dbo in にエイリアスとして指定されている場合は、db1 の “loginA” で作成されたすべてのオブジェクトが dbo によって所有されます。ただし、Adaptive Server はログイン名と作成者のデータベース・ユーザ ID については、オブジェクトの所有権を具体的に記録します。「具体的 ID」(179 ページ) を参照してください。そのデータベース内でオブジェクトを具体的に所有している場合は、データベースからエイリアスを削除することはできません。

注意 データベース内にオブジェクトを作成したログインのエイリアスを削除することはできません。一般に、テーブル、プロシージャ、ビュー、トリガを所有していないユーザについてのみ、エイリアスを使用してください。

エイリアスの追加

ユーザのエイリアスを追加するには、sp_addalias を使用します。

sp_addalias *loginame*, *name_in_db*

各パラメータの意味は、次のとおりです。

- *loginame* には、現在のデータベースにエイリアスを作成するユーザの名前を指定します。Adaptive Server のアカウントを持つユーザでなければなりませんが、現在のデータベースのユーザであってはなりません。
- *name_in_db* には、*loginame* で指定したユーザをリンクするデータベース・ユーザの名前を指定します。*name_in_db* は、現在のデータベース内の sysusers に存在する必要があります。

sp_addalias を実行すると、*loginame* で指定したユーザ名が、*name_in_db* で指定したユーザ名にマップされます。そのために、システム・テーブル sysalternates にローが 1 つ追加されます。

ユーザがデータベースを使用しようとすると、Adaptive Server は、sysusers 内でそのユーザのサーバ・ユーザ ID 番号 (*suid*) を検索します。見つからない場合は、次に sysalternates を調べます。ここでユーザの *suid* が見つかり、データベース・ユーザの *suid* にマップされている場合、最初のユーザは、このデータベースを使用している間は 2 番目のユーザとして扱われます。

たとえば、Mary がデータベースを所有しているとします。Jane と Sarah の 2 人が所有者と同様にこのデータベースを使用できるようにします。Jane と Sarah は Adaptive Server のログインを持っていますが、Mary のデータベースを使用する権限はありません。Mary は次のコマンドを実行します。

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

警告！ データベース所有者としてのエイリアスを与えられたユーザは、そのデータベースに関して、すべてのパーミッションを持ち、データベース所有者が実行できるすべてのアクションを実行できます。データベース所有者は、データベースに対する完全なアクセス権を他のユーザに与えることによって発生する危険性について、十分に検討する必要があります。

エイリアスの削除

代替 *suid* からユーザ ID へのマッピングを削除するには、sp_dropalias を使用します。これによって、sysalternates から関連するローが削除されます。構文は次のとおりです。*loginame* は、sp_addalias で名前をマップしたときに *loginame* として指定されたユーザの名前です。

sp_dropalias *loginame*

エイリアスを削除すると、ユーザはそのデータベースにアクセスできなくなります。

エイリアスを持つログインによって作成されたオブジェクトやスレッショルドがある場合は、そのエイリアスを削除することはできません。これらの操作を実行したエイリアスを sp_dropalias で削除する前に、そのオブジェクトまたはプロシージャを削除してください。エイリアスを削除した後にもそのオブジェクトが必要な場合は、別の所有者で再作成します。

エイリアス情報を取得する方法

エイリアスについての情報を表示するには、sp_helpuser を使います。たとえば、“dbo” のエイリアスを表示するには、次のように実行します。

```
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
```

ユーザ情報を取得する方法

表 3-6 は、ユーザ、グループ、現在の Adaptive Server の使用状況に関する情報を表示するために使用するプロシージャを示します。

表 3-6: Adaptive Server のユーザとグループの情報の表示

タスク	プロシージャ
現在の Adaptive Server のユーザとプロセスのレポート	sp_who
ログイン・アカウントに関する情報の表示	sp_displaylogin
データベース内のユーザとエイリアスのレポート	sp_helpuser
データベース内のグループのレポート	sp_helpgroup

ユーザとプロセスをレポートする方法

sp_who を使用すると、Adaptive Server の現在のユーザとプロセスについての情報が表示されます。

```
sp_who [loginame | "spid"]
```

各パラメータの意味は、次のとおりです。

- *loginame* には、ユーザの Adaptive Server ログイン名を指定します。ログイン名を指定して sp_who を実行すると、そのユーザによって実行されているプロセスについての情報が表示されます。
- *spid* には、特定のプロセスの番号を指定します。

sp_who は、実行中の各プロセスについて、サーバ・プロセス ID のセキュリティ関連情報、ステータス、プロセス・ユーザのログイン名、実際のログイン名 (*login_name* がエイリアスの場合)、ホスト・コンピュータの名前、このプロセスをブロックしているプロセスがある場合はそのサーバ・プロセス ID、データベースの名前、実行中のコマンドをレポートします。

ログイン名も *spid* も指定せずに sp_who を実行した場合は、すべてのユーザが実行しているプロセスについての情報が表示されます。

パラメータを指定しないで sp_who を実行した場合のセキュリティ関連の例を次に示します。

```
fid spid  status      loginame   orignameÂ   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 の出力では、システム・プロセスの *loginame* はすべて NULL です。

ログイン・アカウントに関する情報の取得

指定のログイン・アカウント、またはワイルドカードのパターンと一致するログイン名に関する、付与されたすべての役割などの情報を表示するには、sp_displaylogin を使用します。*loginame* (またはパターン一致のワイルドカード) は、情報が必要なユーザ・ログイン名のパターンです。

sp_displaylogin [*loginame* | *wildcard*]

システム・セキュリティ担当者でもシステム管理者でもないユーザは、自分のアカウントに関する情報だけを取得できます。システム・セキュリティ担当者またはシステム管理者の場合は、*loginame* | *wildcard* パラメータを使用して、すべてのアカウントに関する情報にアクセスできます。

sp_displaylogin は、使用しているサーバ・ユーザ ID、ログイン名、フルネーム、各自に付与されたすべての役割、最後のパスワード変更日付、デフォルト・データベース、デフォルト言語、使用しているアカウントがロックされているかどうか、自動ログイン・スクリプト、パスワード有効期間、パスワードの有効期間が切れたかどうか、ログインに使用されたパスワード暗号化のバージョン、およびログインに指定された認証メカニズムを表示します。

sp_displaylogin は、ユーザに付与されている役割をすべて表示するので、set コマンドで無効にされている役割であっても表示されます。たとえば、次に sa の役割を表示します。

```
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)
```

データベース・ユーザ情報を取得する方法

現在のデータベースを使用する権限を与えられているユーザについての情報を表示するには、sp_helpuser を使用します。*name_in_db* は、現在のデータベースのユーザ名です。

sp_helpuser [*name_in_db*]

ユーザ名を指定して sp_helpuser を実行すると、そのユーザについての情報が表示されます。ユーザ名を指定しない場合は、すべてのユーザについての情報が表示されます。

次の例では、データベース pubs2 で、パラメータを指定しないで sp_helpuser を実行した結果を示します。

```
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
```

ユーザの名前と ID を表示する方法

ユーザのサーバ・ユーザ ID またはログイン名を表示するには、suser_id と suser_name を使用します。

表 3-7: suser_id システム関数と suser_name システム関数

表示対象	使用	指定する引数
サーバ・ユーザ ID	suser_id	(["server_user_name"])
サーバ・ユーザ名 (ログイン名)	suser_name	([server_user_ID])

これらのシステム関数の引数は省略可能です。引数を指定しない場合は、現在のユーザの情報が表示されます。

次の例では、ユーザ “sandy” のサーバ・ユーザ ID が表示されます。

```
select suser_id("sandy")

------
     3
```

次の例は、“mary” というログイン名のシステム管理者が、引数を指定しないでコマンドを実行する方法を示します。

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

データベース内のユーザの ID 番号や名前を表示するには、user_id と user_name を使用します。

表 3-8: user_id システム関数と user_name システム関数

表示対象	使用	指定する引数
ユーザ ID	user_id	([db_user_name])
ユーザ名	user_name	([db_user_ID])

これらのシステム関数の引数は省略可能です。引数を指定しない場合は、現在のユーザの情報が表示されます。次に例を示します。

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

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

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

ユーザ情報の変更

表 3-9 は、パスワード、デフォルト・データベース、デフォルト言語、フルネーム、グループの割り当ての変更に使用するシステム・プロシージャを示します。

表 3-9: ユーザ情報を変更するためのコマンドまたはシステム・プロシージャ

タスク	必要な役割	システム・プロシージャ –	ログイン/ログイン・プロファイルの 変更/作成/削除コマンドの対象となるマスタ・データベース
パスワードの変更	ユーザ	alter login	任意のデータベース
他のユーザのパスワードの変更	システム・セキュリティ担当者	alter login	任意のデータベース
認証メカニズムの変更	システム・セキュリティ担当者	alter login alter login profile	任意のデータベース
フルネームの変更	システム・セキュリティ担当者	alter login	任意のデータベース
独自のフルネームの変更	ユーザ	alter login	任意のデータベース
デフォルト言語またはデフォルト・データベースの変更	システム・セキュリティ担当者	alter login profile alter login	任意のデータベース
ユーザのグループの割り当ての変更	システム管理者、データベース所有者、またはシステム・セキュリティ担当者	sp_changegroup	ユーザ・データベース
ログイン・プロファイルの変更	システム・セキュリティ担当者	alter login profile	任意のデータベース
ログイン・トリガの設定	システム・セキュリティ担当者	alter login profile	任意のデータベース

パスワードの変更

alter login を使用すると、すべてのユーザがいつでも自分のパスワードを変更できます。システム・セキュリティ担当者は、alter login を使用して、他のユーザのパスワードを変更できます。

たとえば、ron という名前のログイン・アカウントのパスワードを変更するには、次を入力します。

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

『リファレンス・マニュアル：コマンド』の「alter login」を参照してください。

新しいパスワードの要求

systemwide password expiration 設定パラメータを使用して、パスワードの有効期間を設定できます。これは、すべての Adaptive Server ユーザに対して、各自のパスワードを定期的に変更するよう強制的に指示するものです。『システム管理ガイド 第 1 巻』の「第 5 章 設定パラメータ」を参照してください。systemwide password expiration を使用しない場合でも、セキュリティ上の理由から、ユーザが各自のパスワードを定期的に変更することは重要です。

設定パラメータは、パスワード・ポリシー設定に置き換えられます。

password expiration interval は、パスワード有効期限の間隔を日数で指定します。0 ～ 32767 の任意の値を指定できます。たとえば、パスワードの有効期限の間隔が 30 日である新しいログオンを 2007 年 8 月 1 日の午前 10 時半に作成したとすると、2007 年 8 月 31 日の午前 10 時半にパスワードの有効期限が切れます。

syslogins テーブルのカラム pwdate には、パスワードが最後に変更された日が記録されています。次のクエリは、2007 年 9 月 15 日以降パスワードが変更されていないすべてのログイン名を選択します。

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

null パスワード

null パスワードを割り当てることはできません。ただし、Adaptive Server がインストールされるとき、デフォルトの “sa” アカウントのパスワードは null に設定されます。次に null パスワードを有効なパスワードに変更する方法の例を示します。

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

注意 文の中で “null” を引用符で囲まないでください。

パスワードが失われた場合のログイン

次のような状況が発生する場合は、dataserver -p*login_name* を使用してください。

- システム管理者のログイン・アカウントがすべてロックされている。
- システム・セキュリティ担当者のログイン・アカウントがすべてロックされている。
- sa_role または sso_role のパスワードが失われた。

そのような場合は、dataserver パラメータを –p パラメータと一緒に使用すると、上記のアカウントと役割の新しいパスワードを設定できます。*login_name* は、パスワードを再設定する必要があるユーザの名前または役割の名前 (sa_role または sso_role) です。

-p パラメータを使用して起動すると、Adaptive Server は、ランダムなパスワードを生成、表示、暗号化してから、そのアカウントまたは役割の新しいパスワードとして master..syslogins または master..syssrvroles に保存します。

サーバの再起動時に、パスワードを変更することを強くおすすめします。たとえば、sa_role を持つユーザ rsmith のパスワードを再設定するには、次のように入力します。

```
dataserver -prsmith
```

sso_role のパスワードを再設定するには、次のように入力します。

```
dataserver -psso_role
```

ユーザ・セッション情報の変更

set コマンドには、各クライアントに個別の名前、ホスト名、アプリケーション名を割り当てるオプションがあります。これは、Adaptive Server に多数のクライアントが同じ名前、ホスト名、またはアプリケーション名を使用して接続するシステムにおいてクライアントを区別するのに便利です。

以下は、set コマンドの構文の一部です。

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

各パラメータの意味は、次のとおりです。

- *client_name* − クライアントに割り当てる名前です。
- *host_name* − クライアントの接続元のホスト名です。
- *application_name* − Adaptive Server に接続しているアプリケーションです。

これらのパラメータは、sysprocesses テーブルのカラム clientname、clienthostname、clientapplname に格納されます。

たとえば、ユーザが Adaptive Server に “client1” としてログインする場合、次のようなコマンドを使用して、個々のクライアントの名前、ホスト名、アプリケーション名を割り当てます。

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

このユーザは、ホスト “money1” から “webserver2” アプリケーションを使用してログインするユーザ “alison” として sysprocesses テーブルに登録されます。ただし、新しい名前は sysprocesses に登録されていてもパーミッションの検査には使用されず、sp_who を実行すると、このクライアント接続は元のログイン (上の例の場合は client1) に属しているとして表示されます。set clientname を実行しても、set proxy とは異なり、他のユーザのパーミッション、ログイン名、*suid* を使用できるようにはなりません。

設定できるのは、自分の現在のクライアント・セッションのクライアント名、ホスト名、アプリケーション名だけです (ただし、表示はどのクライアント接続であっても可能です)。また、ユーザがログアウトすると、この情報は消滅します。これらのパラメータは、ユーザがログインするたびに割り当て直す必要があります。たとえば、ユーザ “alison” は、他のクライアント接続のクライアント名、ホスト名、アプリケーション名を設定することはできません。

クライアントの接続情報を表示するには、そのクライアントのシステム・プロセス ID を使用します。たとえば、上記の例のユーザ “alison” が *spid* 13 で接続しているときに、次のコマンドを発行すると、このユーザのすべての接続情報が表示されます。

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

現在のクライアント接続情報を表示するには (たとえば、ユーザ “alison” が自分の接続情報を表示する場合)、次のように入力します。

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

ユーザおよびグループの削除

システム管理者、システム・セキュリティ担当者、またはデータベース所有者は、sp_dropuser または sp_dropgroup を使用して、ユーザとグループをデータベースから削除します。

ユーザの削除

データベース所有者、システム・セキュリティ担当者、またはシステム管理者は、sp_dropuser を使用して Adaptive Server ユーザがデータベースにアクセスできないようにすることができます。その場合は、そのデータベース内で sp_dropuser を実行します。(“guest” ユーザがそのデータベースに定義されている場合、ユーザは引き続きそのデータベースに対して “guest” としてアクセスできます)。

構文は次のとおりです。別の名前が sp_adduser を使用して割り当てられていなければ、*name_in_db* は通常はログイン名です。

sp_dropuser *name_in_db*

オブジェクトを所有しているユーザを削除することはできません。オブジェクトの所有権を譲渡するコマンドはないので、そのユーザが所有しているオブジェクトを削除してから、ユーザを削除してください。オブジェクトを所有しているユーザのアクセスを禁止するには、sp_locklogin を使用して、そのユーザのアカウントをロックします。

別のユーザにパーミッションを付与しているユーザも削除できません。revoke with cascade を使い、パーミッションを付与されているすべてのユーザからパーミッションを取り消した後で、ユーザを削除します。その後で、必要に応じてユーザにパーミッションを付与し直してください。

グループの削除

システム・セキュリティ担当者、システム管理者、またはデータベース管理者は、sp_dropgroup を使用してグループを削除します。構文は次のとおりです。

```
sp_dropgroup grpname
```

メンバを持っているグループを削除することはできません。削除しようとすると、そのグループのメンバの一覧を示すエラー・メッセージが表示されます。グループからユーザを削除するには、sp_changegroup を使用します。詳細については、「ユーザのグループ・メンバシップの変更」(69 ページ) を参照してください。

ライセンス使用状況のモニタリング

License Use Monitor を使用すると、システム管理者は Adaptive Server で使用されているユーザ・ライセンスの数をモニタリングし、ライセンス契約のデータの管理を安全に行うことができます。つまり、Adaptive Server で使用されているライセンスの数が、ライセンス契約で指定されている数を超えないようにすることができます。

License Use Monitor は発行されたライセンスの数を追跡しますが、ライセンス契約を強制することはありません。ライセンス契約で指定された数を超えてユーザ・ライセンスを使用していると License Use Monitor が通知した場合は、担当の Sybase 販売代理店にお問い合わせください。

License Use Monitor を設定するには、システム管理者の権限が必要です。デフォルトでは、Adaptive Server がインストールまたはアップグレードされた直後は、モニタはオフになっています。

以下の 「License Use Monitor の設定」を参照してください。

ライセンスがカウントされる仕組み

ライセンスは、ホスト・コンピュータ名とユーザ名の組み合わせとなります。あるユーザが Adaptive Server に同じホスト・マシンから 2 回以上ログインしても、1 ライセンスが使用されます。しかし、そのユーザがホスト A から 1 回、ホスト B から 1 回ログインすると、2 ライセンスが使用されます。複数のユーザが同じホストからそれぞれ異なるユーザ名で Adaptive Server にログインした場合、個々のユーザ名とホスト名の組み合わせが 1 ライセンスを使用します。

License Use Monitor の設定

sp_configure を使用して、ライセンス契約で定められたライセンス数を指定します。*number* はライセンス数です。

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

この例ではユーザ・ライセンスの最大数を 300 に設定するので、ライセンス番号が 301 になるとライセンス数を超えていることがレポートされます。

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

ユーザ・ライセンス数を増やした場合は、license information 設定パラメータも変更する必要があります。

ハウスキーピング・タスクを使用したライセンス使用状況のモニタリング

License Use Monitor が設定されると、ハウスキーピング・タスクは、Adaptive Server にログインしている各ユーザのユーザ ID とホスト名を基に使用されているユーザ・ライセンスの数を調べます。License Use Monitor は、使用中のユーザ・ライセンスの最大数を記録する変数を更新します。

- 使用中のライセンス数が、前回のハウスキーピング実行時と同じかそれよりも減っている場合は、License Use Monitor は何も処理を実行しません。
- 使用中のライセンス数が、前回のハウスキーピング実行時よりも増えている場合は、License Use Monitor はこの数を使用中のライセンスの最大数として設定します。
- 使用中のライセンス数がライセンス契約に定められた数より多い場合、License Use Monitor はエラー・ログに次のようなメッセージを発行します。

  ```
  Exceeded license usage limit.Contact Sybase Sales for
  additional licenses.
  ```

ハウスキーピング・チョア・タスクは、Adaptive Server のアイドル・サイクル中に実行されます。License Use Monitor がライセンスの使用状況を追跡するようにするには、housekeeper free write percent と license information 設定パラメータを 1 以上に設定します。

ハウスキーピング・チョア・タスクの詳細については、『パフォーマンス＆チューニング・シリーズ：基本』の「第 3 章 エンジンと CPU の使用方法」を参照してください。

ユーザ・ライセンス数のロギング

Adaptive Server をインストールまたはアップグレードするときに、master データベース内に syblicenseslog システム・テーブルが作成されます。表 3-10 に示すように、License Use Monitor は 24 時間ごとに、syblicenseslog 内のカラムを更新します。

表 3-10: syblicenseslog テーブル内のカラム

カラム	説明
status	-1 － ハウスキーピング機能によるライセンス数のモニタはできない 0 － ライセンス数は制限を超過していない 1 － ライセンス数は制限を超過している
logtime	ログ情報が挿入された日付と時刻
maxlicenses	24 時間の間に使用されたライセンス数の最大数

次は、syblicenseslog の例です。

```
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
```

この例では、1998 年 7 月 19 日に使用中のユーザ・ライセンス数が制限を超えています。

Adaptive Server が停止すると、License Use Monitor は現在の最大使用ライセンス数を使用して syblicenseslog を更新します。Adaptive Server が再起動すると、新たな 24 時間のモニタリング期間が開始します。

1998 年 7 月 21 日の 2 番目のローは、サーバの停止と再起動によって挿入されたものです。

ユーザ ID とログイン ID の番号

Adaptive Server でサポート可能なサーバ当たりのログイン数とデータベース当たりのユーザ数は 20 億を超えます。ID に使用可能な番号の範囲を広げるために、正の値だけでなく負の値も使用されます。

ID 番号の制限と範囲

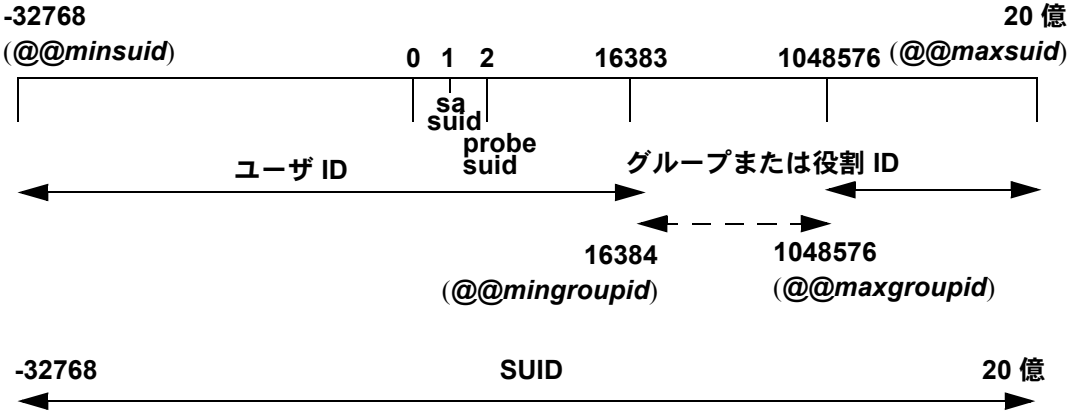
表 3-11 は、ID タイプごとの有効な範囲を示します。

表 3-11: ID タイプの範囲

ID タイプ	サーバの制限
サーバ当たりのログイン数 (*suid*)	20 億に 32K を加えた数
データベース当たりのユーザ数 (*uid*)	20 億から 1032193 を引いた数
データベース当たりのグループまたは役割の数 (*gid*)	16,384 ～ 1,048,576

図 3-1 は、ログイン、ユーザ、グループの制限と範囲を示します。

図 3-1: Adaptive Server で使用できるユーザ、グループ、ログイン



ユーザ ID (*uid*) に負の値が使用されることがあります。

sysusers でグループや役割に割り当てられているサーバ・ユーザ ID (*suid*) の値は、ユーザ ID (*uid*) の符号を逆にした値であるとは限りません。*sysusers* でグループや役割に関連付けられている suid はすべて、-2 (INVALID_SUID) に設定されます。

ログイン接続の制限

Adaptive Server ではサーバごとに 20 億以上のログインを定義できますが、実際に Adaptive Server への同時接続が可能なユーザの数は、次に示す値によって制限されます。

- number of user connections 設定パラメータの値
- Adaptive Server で使用できるファイル記述子の数 (各ログインは接続ごとにファイル記述子を 1 つ使用する)

注意　サーバ上で同時に実行されるタスクの最大数は 32,000 です。

❖ ログインと同時接続の数を最大にする

1　Adaptive Server が実行されるオペレーティング・システムを、32,000 個以上のファイル記述子を使用できるように設定します。

2　number of user connections の値を 32,000 以上に設定します。

注意　Adaptive Server で 64K を超える数のログインと同時接続を可能にするには、最初に、64K を超えるファイル記述子を使用できるようにオペレーティング・システムを設定する必要があります。ファイル記述子数を増やす方法については、オペレーティング・システムのマニュアルを参照してください。

表 3-12: ログイン、ユーザ、グループに関するグローバル変数

変数名	表示対象	値
@@invaliduserid	無効ユーザ ID	-1
@@minuserid	最小のユーザ ID	-32768
@@guestuserid	guest ユーザ ID	2
@@mingroupid	最小のグループまたは役割ユーザ ID	16384
@@maxgroupid	最大のグループまたは役割ユーザ ID	1048576
@@maxuserid	最大のユーザ ID	2147483647
@@minsuid	最小のサーバ・ユーザ ID	-32768
@@probesuid	プローブ・サーバ・ユーザ ID	2
@@maxsuid	最大のサーバ・ユーザ ID	2147483647

グローバル変数を表示するには、次のように入力します。

```
select variable_name
```

次に例を示します。

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

使用状況に関する情報の表示：チャージバック・アカウンティング

ユーザが Adaptive Server にログインすると、そのユーザの CPU と I/O の使用量の累積が始まります。Adaptive Server は、1 人のユーザまたはすべてのユーザの合計使用量をレポートできます。各ユーザの情報は master データベース内の syslogins システム・テーブルに保存されます。

現在使用量の統計のレポート

システム管理者は、sp_reportstats または sp_clearstats を使用して、Adaptive Server 上の個々のユーザまたはすべてのユーザの現在の合計使用量のデータを表示したりクリアしたりすることができます。

現在のアカウンティング合計の表示

sp_reportstats は、Adaptive Server ユーザの現在の合計使用量を表示します。CPU および I/O の合計使用量と、これらのリソースの使用率を表示します。“sa” ログイン・アカウント (*suid* が 1 のプロセス)、チェックポイント、ネットワーク、ミラー・ハンドラについての統計は記録されません。

新しいアカウンティング期間の開始

sp_clearstats を実行して syslogins の合計値をクリアするまで、Adaptive Server の CPU と I/O の統計が累積されます。sp_clearstats を実行すると、Adaptive Server ユーザについての統計の新しい累積期間が開始し、sp_reportstats が実行されて前回の累積期間の統計が出力されます。

アカウンティング期間の長さは、各サイトでの統計の使用方法に従って選択してください。たとえば、Adaptive Server の CPU と I/O の使用率に応じて、月ごとに各部門へのアカウンティングを行う場合は、月に一度 sp_clearstats を実行します。

これらのストアド・プロシージャの詳細については、『リファレンス・マニュアル：プロシージャ』を参照してください。

アカウンティング統計を追加する間隔の指定

システム管理者は、設定パラメータを使用して、課金統計を syslogins に追加する頻度を指定できます。

課金統計を syslogins に追加する基準となるマシンの累積クロック・チック数を指定するには、cpu accounting flush interval 設定パラメータを使用します。デフォルト値は 200 です。次に例を示します。

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

システムの 1 チックの長さ (マイクロ秒単位) を調べるには、Adaptive Server で次のクエリを実行します。

```
select @@timeticks
```

情報を syslogins に追加 (フラッシュ) する基準となる読み込みまたは書き込み I/O の累積数を指定するには、i/o accounting flush interval 設定パラメータを使用します。デフォルト値は 1000 です。次に例を示します。

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

I/O と CPU 統計は、ユーザの I/O または CPU の累積使用量が指定値を超えるとフラッシュされます。ユーザが Adaptive Server のセッションを終了したときも、情報はフラッシュされます。

どちらの設定パラメータも、最小値は 1、最大値は 2,147,483,647 です。

第 4 章

外部認証

この章では、Adaptive Server の外部のレポジトリに保管されている認証データを使用してユーザを認証する Adaptive Server の機能について説明します。

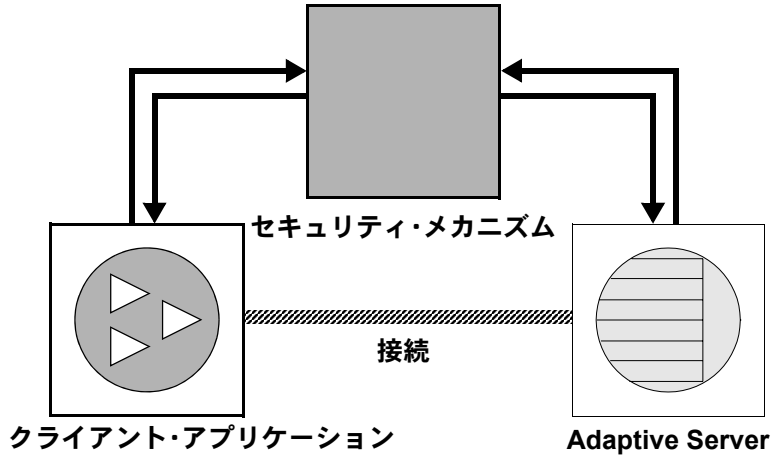
大規模な異機種アプリケーションでは、ログインを集中レポジトリで認証することによってセキュリティを強化できます。Adaptive Server では、次の外部認証メソッドがサポートされています。

- Kerberos － インフラストラクチャを使用するエンタープライズ環境において、集中化された安全な認証メカニズムを提供します。KDC (Key Distribution Center) と呼ばれる信頼されたサード・パーティのサーバを使用して認証が行われ、クライアントとサーバの両方が検証されます。
- LDAP ユーザ認証 － LDAP (Lightweight Directory Access Protocol) は、ユーザのログイン名とパスワードに基づく集中化された認証メカニズムを提供します。
- PAM ユーザ認証 － PAM (Pluggable Authentication Module) は、管理インタフェースおよびランタイム・アプリケーション・インタフェースとしてオペレーティング・システムが提供するインタフェースを使用した、集中化された認証メカニズムを提供します。

ネットワークベース・セキュリティでの Adaptive Server の設定

クライアントとサーバの間のセキュア接続は、ログイン認証とメッセージ保護のために使用できます。

図 4-1: クライアントと Adaptive Server の間のセキュア接続の確立



クライアントが認証サービスを要求するときは、次の手順に従います。

1 クライアントは、セキュリティ・メカニズムを使用してログインを検証します。セキュリティ・メカニズムから、セキュリティ関連情報が格納されたクレデンシャルが返されます。

2 クライアントは、Adaptive Server にクレデンシャルを送信します。

3 Adaptive Server は、セキュリティ・メカニズムを使用してクライアントのクレデンシャルを認証します。クレデンシャルが有効な場合は、クライアントと Adaptive Server の間にセキュア接続が確立されます。

クライアントがメッセージ保護サービスを要求するときは、次の手順に従います。

1 クライアントは、セキュリティ・メカニズムを使用して、Adaptive Server に送るデータ・パケットを準備します。

   セキュリティ・メカニズムは、要求されるセキュリティ・サービスに応じて、データを暗号化するか、またはデータと対応する暗号化シグニチャを作成します。

2 クライアントは、Adaptive Server にデータ・パケットを送信します。

3 Adaptive Server は、データ・パケットを受信すると、セキュリティ・メカニズムを使用して復号化と検証を行います。

4 Adaptive Server は、結果をクライアントに返します。このとき、セキュリティ・メカニズムを使用して、要求されたセキュリティ機能を実行します。たとえば、暗号化された形式で結果を返します。

セキュリティ・サービスと Adaptive Server

選択したセキュリティ・メカニズムに応じて、次のセキュリティ・サービスを使用できます。

- 統一化ログイン － ユーザを一度だけ認証する。ユーザは、Adaptive Server にログインするたびに名前とパスワードを入力する必要はない。
- メッセージの機密保持 － ネットワーク上で転送されるデータを暗号化する。
- 相互認証 － クライアントとサーバの身元を検証する。相互認証を要求できるのはクライアントのみです。Adaptive Server は相互認証を要求できません。
- メッセージ整合性 － データ通信が変更されていないことを検証する。
- リプレイの検出 － データが侵入者によって傍受されていないことを確認する。
- 順序不整合の検査 － データ通信の順序を確認する。
- メッセージ・オリジンの検査 － メッセージのオリジンを確認する。
- クレデンシャルの委任 － クライアントが Adaptive Server にクレデンシャルを委任して、リモート・サーバと安全に接続できるようにする。このサービスは Kerberos セキュリティ・メカニズムによってサポートされる。Adaptive Server は、現時点では、CIS からのリモート Adaptive Server への接続でこのサービスをサポートする。
- リモート・プロシージャ・セキュリティ － Kerberos 接続の CIS を介したリモート・プロシージャ通信での相互認証、メッセージの機密性、メッセージの整合性を保証する。

注意 使用するセキュリティ・メカニズムで、これらのサービスすべてを利用できるとは限りません。詳細については、「使用できるセキュリティ・サービスの情報の取得」(105 ページ) を参照してください。

ネットワークベース・セキュリティの管理

表 4-1 は、Adaptive Server のネットワークベース・セキュリティ機能を使用するための全体的なプロセスを示します。Adaptive Server をインストールしてから、表 4-1 の手順を実行します。

表 4-1: ネットワークベース・セキュリティの管理

手順	説明	参照箇所
1. 次の設定ファイルを設定する。 • *libtcl.cfg* • *objectid.dat* • *interfaces* (またはディレクトリ・サービス)	*libtcl.cfg* ファイルを編集する。 *objectid.dat* ファイルを編集する。 *interfaces* ファイルまたはディレクトリ・サービスを編集する。	• 「セキュリティの設定ファイルの設定」(93 ページ) • 使用しているプラットフォームの『Open Client/Server 設定ガイド』
2. セキュリティ・メカニズムのセキュリティ管理者によって各ユーザおよび Adaptive Server と Backup Server 用のログインが作成されていることを確認する。	セキュリティ管理者は、ユーザとサーバの名前とパスワードをセキュリティ・メカニズムに追加する必要がある。	• 使用しているセキュリティ・メカニズムのマニュアル • 「セキュリティ・メカニズムに対するユーザとサーバの識別」(97 ページ)
3. インストール環境にセキュリティを設定する。	sp_configure を使用する。	「Adaptive Server でのセキュリティの設定」(98 ページ)
4. Adaptive Server を再起動します。	use security services パラメータをアクティブにする。	使用しているプラットフォームの『設定ガイド』
5. 企業全体のログインをサポートするためのログインを Adaptive Server に追加する。	create login を使用してログインアカウントを追加する。オプションで、sp_configure にデフォルト・セキュア・ログインを指定する。	「統一化ログインをサポートするためのログインの追加」(101 ページ)
6. リモート・サーバに必要なセキュリティ・メカニズムを有効にする。	sp_serveroption の security mechanism オプションを使用して、リモート・サーバに必要なセキュリティ・メカニズムを有効にする。	「リモート接続での Kerberos セキュリティの確立」(103 ページ)
7. サーバに接続し、セキュリティ・サービスを使用する。	isql_r や Open Client Client-Library を使い、使用するセキュリティ・サービスを指定して、Adaptive Server に接続する。	• 「サーバへの接続とセキュリティ・サービスの使用」(104 ページ) • 使用しているプラットフォームの『Open Client/Server 設定ガイド』 • 『Open Client Client-Library/C リファレンス・マニュアル』の「セキュリティ機能」
8. 利用できるセキュリティ・サービスとセキュリティ・メカニズムをチェックする。	show_sec_services 関数および is_sec_services_on 関数を使用し、利用できるセキュリティ・サービスをチェックする。 Adaptive Server がサポートしているセキュリティ・メカニズムとそのセキュリティ・サービスのリストについては、select を使用して syssecmechs システム・テーブルを問い合わせる。	「使用できるセキュリティ・サービスの情報の取得」(105 ページ)

セキュリティの設定ファイルの設定

設定ファイルは、インストール時に Sybase ディレクトリ構造内のデフォルト・ロケーションに作成されます。

表 4-2: 設定ファイルの名前とロケーション

ファイル名	説明	ロケーション
libtcl.cfg	このドライバ設定ファイルには、ディレクトリ、セキュリティ、ネットワークの各ドライバに関する情報と、必要な初期化情報が格納されている。	*UNIX* プラットフォーム：*$SYBASE/$SYBASE_OCS/config* *Windows* プラットフォーム：*%SYBASE%\%SYBASE_OCS%\ini*
objectid.dat	オブジェクト識別子ファイルは、文字セット、照合順、セキュリティ・メカニズムのロケール名にグローバル・オブジェクト識別子をマップする。	*UNIX* プラットフォーム：*$SYBASE/config* *Windows* プラットフォーム：*%SYBASE%\ini*
UNIX：*interfaces* デスクトップ・プラットフォーム：*sql.ini*	*interfaces* ファイルには、ファイルにリストされている各サーバの接続とセキュリティ情報が含まれる。 **注意**　Adaptive Server version 12.5.1 以降では、*interfaces* ファイルの代わりにディレクトリ・サービスを使用できる。	*UNIX* プラットフォーム：*$SYBASE* デスクトップ・プラットフォーム：*SYBASE_home\ini*

設定ファイルの詳細な説明については、使用しているプラットフォームの『Open Client/Server 設定ガイド』を参照してください。

サーバのセキュリティ情報の指定

インストール環境のサーバに関する情報を定義するには、*interfaces* ファイルまたはディレクトリ・サービスを使用します。

interfaces ファイルには、サーバのネットワークおよびセキュリティの情報が格納されています。セキュリティ・サービスを使用するには、そのセキュリティ・サービスのグローバル識別子を指定する “secmech” 行を *interfaces* ファイルに追加する必要があります。

Adaptive Server では、サーバに関する情報を記録するディレクトリ・サービスを使用できます。ディレクトリ・サービスは、ネットワーク・サーバに関する情報の作成、修正、検索を管理します。ディレクトリ・サービスを使用する利点は、新しいサーバがネットワークに追加されたときやサーバのアドレスが変更されたときに複数の *interfaces* ファイルを更新しなくて済むことです。ディレクトリ・サービスとともにセキュリティ・サービスを使用するには、そのセキュリティ・サービスのグローバル識別子を 1 つ以上指定するように、secmech セキュリティ属性を定義する必要があります。

セキュリティ・メカニズムを指定する UNIX ツール

使用するセキュリティ・メカニズムは、次のように指定します。

- *interfaces* ファイルを使用する場合は、dscp ユーティリティを使用する。
- ディレクトリ・サービスを使用する場合は、dscp_r ユーティリティを使用する。

注意 *interfaces* ファイルまたはディレクトリ・サービスのエントリを作成するのに役立つ dsedit ツールを、UNIX プラットフォームで利用できます。ただし、このツールでは、セキュリティ・メカニズムの secmech エントリを作成することはできません。

dscp の詳細については、『Open Client/Server 設定ガイド UNIX 版』を参照してください。

サーバの属性を指定するデスクトップ・ツール

sql.ini ファイルまたはディレクトリ・サービスで、システムのサーバに関する情報を指定するには、dsedit ユーティリティを使用します。このユーティリティのグラフィカル・ユーザ・インタフェースを使うと、サーバのバージョン、名前、セキュリティ・メカニズムなどのサーバ属性を指定できます。セキュリティ・メカニズムの属性については、使用する予定のセキュリティ・メカニズムに対応するオブジェクト識別子を指定できます。dsedit の使用方法については、『Open Client/Server 設定ガイド デスクトップ・プラットフォーム版』を参照してください。

ネットワーク・ベース・セキュリティを使用するための *libtcl.cfg* の準備

libtcl.cfg と *libtcl64.cfg* (64 ビット・アプリケーション用) には、以下の 3 種類のドライバに関する情報が含まれます。

- ネットワーク (Net-Library)
- ディレクトリ・サービス
- セキュリティ

「ドライバ」は、外部サービス・プロバイダとのインタフェースとなる Sybase ライブラリです。ドライバは動的にロードされるため、アプリケーションが使用するドライバを変更しても、アプリケーションの再リンクは必要ありません。

ネットワーク・ドライバのエントリ

ネットワーク・ドライバ・エントリの構文は、次のとおりです。

driver=protocol description

各要素の意味は次のとおりです。

- *driver* は、ネットワーク・ドライバの名前です。
- *protocol* は、ネットワーク・プロトコルの名前です。

- *description* は、エントリの説明です。この要素はオプションです。

注意 ネットワーク・ドライバを指定しない場合は、アプリケーションとプラットフォームに適したドライバが自動的に選択されます。たとえば、UNIX プラットフォームでは、セキュリティ・サービスが使用されるときに、スレッドを処理できるドライバが自動的に選択されます。

ディレクトリ・サービスのエントリ

interfaces ファイルの代わりにディレクトリ・サービスを使用する場合は、ディレクトリ・サービスのエントリが適用されます。使用しているプラットフォームの設定ガイドおよび『Open Client/Server 設定ガイド』を参照してください。

セキュリティ・ドライバのエントリ

セキュリティ・ドライバ・エントリの構文は次のとおりです。

```
provider=driver init-string
```

各要素の意味は次のとおりです。

- *provider* － セキュリティ・メカニズムのローカル名。ローカル名からグローバル・オブジェクト識別子へのマッピングは、*objectid.dat* で定義される。

  デフォルトのローカル名は次のとおりです。

  - “csfkrb5” － CyberSAFE Kerberos または MIT Kerberos セキュリティ・メカニズム用
  - “LIBSMSSP” － Windows NT または Windows 95 (クライアントのみ) の Windows LAN Manager 用

  デフォルト以外のローカル・メカニズム名を使用する場合は、objectid.dat ファイルにあるローカル名を変更します (「objectid.dat ファイル」(97 ページ) の例を参照)。

- *driver* － セキュリティ・ドライバの名前。UNIX プラットフォームのすべてのドライバのデフォルト・ロケーションは、*$SYBASE/$SYBASE_OCS/lib*。Windows プラットフォームのデフォルト・ロケーションは、*%SYBASE%\%SYBASE_OCS%\dll*。

- *init-string* － ドライバの初期化文字列。この要素はオプションです。*init-string* の値はドライバによって異なる。

  - Kerberos ドライバの場合の *init-string* の構文は次のとおり。ただし、*realm* はデフォルトの Kerberos レルム名。

    ```
    secbase=@realm
    ```

  - Windows NT LAN Manager の場合は、*init-string* は適用されない。

UNIX プラットフォーム情報

libtcl.cfg ファイルを編集する特別なツールはありません。Adaptive Server をインストールした後で、既に存在するエントリをコメント行にしたり、コメントを解除したりするには、通常のエディタを使用します。

Adaptive Server を UNIX プラットフォームにインストールすると、*libtcl.cfg* ファイルの以下の 3 つのセクションにはエントリが既に含まれています。

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

これらのセクションを特定の順序に並べる必要はありません。

使用しないエントリには必ずコメントのマークを付け (先頭に ";" を付ける)、使用するエントリにはコメントのマークを付けない (";" を先頭に付けない) ようにします。

詳細については、『Open Client/Server 設定ガイド UNIX 版』を参照してください。

Sun Solaris の *libtcl.cfg* の例

```
[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
```

[DIRECTORY] セクションのすべてのエントリがコメント行であるため、このファイルはディレクトリ・サービスを使用しません。

ネットワーク・ドライバの [DRIVERS] セクションにあるすべてのエントリもコメント行であるため、適切なドライバがシステムによって自動的に選択されます。セキュリティ・サービスが使用されるときはスレッド・ドライバが自動的に選択され、スレッド・ドライバと連動しないアプリケーションの場合は非スレッド・ドライバが自動的に選択されます。たとえば、Backup Server はセキュリティ・サービスをサポートせず、スレッド・ドライバとは連動しません。

デスクトップ・プラットフォーム情報

ocscfg ユーティリティは、*libtcl.cfg* ファイルのセクションの見出しを自動的に作成します。osccfg を使用した *libtcl.cfg* ファイルの編集もできます。

これは、デスクトップ・プラットフォームの *libtcl.cfg* ファイルの例です。

```
[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
```

『Open Client/Server 設定ガイド デスクトップ・プラットフォーム版』を参照してください。

objectid.dat ファイル

objectid.dat ファイルは、Kerberos サービスを表す 1.3.6.1.4.1.897.4.6.6 などのグローバル・オブジェクト識別子を “csfkrb5” などのローカル名にマッピングします。objectid.dat ファイルには、文字セット用の [CHARSET] セクションや、セキュリティ・サービス用の [SECURITY] セクションが含まれています。objectid.dat ファイルの例を次に示します。

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

libtcl.cfg ファイルでセキュリティ・サービスのローカル名を変更した場合にのみ、テキスト・エディタを使用してこのファイルを変更します。

たとえば、libtcl.cfg に次のセクションがあるとします。

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

上記を次のように変更します。

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

この変更を反映するために、*libtcl.cfg* で objectid.dat を変更します。objectid.dat の Kerberos の行にあるローカル名を次のように変更します。

```
1.3.6.1.4.1.897.4.6.6 = csfkrb5_group
```

注意 セキュリティ・メカニズムごとにローカル名を 1 つだけ指定できます。

セキュリティ・メカニズムに対するユーザとサーバの識別

セキュリティ・メカニズムのセキュリティ管理者は、セキュリティ・メカニズムに対して「プリンシパル」を定義する必要があります。これには、ユーザとサーバの両方が含まれます。次に、ユーザとサーバの追加に使用できるツールを示します。

- Kerberos － ユーザとサーバの定義方法については、Kerberos のベンダ固有のツールを参照。Kerberos と Adaptive Server の詳細については、「Kerberos の使用」(107 ページ) を参照。
- Windows NT LAN Manager － [ユーザー マネージャ] ツールを実行し、Windows NT LAN Manager にユーザを定義する。Adaptive Server の名前を Windows NT LAN Manager にユーザとして定義し、Adaptive Server をそのユーザ名で表示する。

注意 運用環境では、サーバとユーザのキーが含まれているファイルへのアクセスを制御してください。ユーザがこれらのキーにアクセスできる場合は、運用サーバを偽装するサーバをユーザが作成することもできてしまいます。

必要な管理タスクを実行する方法の詳細については、セキュリティ・メカニズムのサード・パーティ・プロバイダのマニュアルを参照してください。

Adaptive Server でのセキュリティの設定

Adaptive Server では、いくつかの設定パラメータを通してネットワークベース・セキュリティを管理します。これらのパラメータを設定するには、システム・セキュリティ担当者の権限が必要です。ネットワークベース・セキュリティに関するパラメータは、すべて「セキュリティ関連」の設定パラメータ・グループに属しています。

ネットワークベース・セキュリティの有効化

ネットワーク-ベース・セキュリティを有効または無効にするには、sp_configure を使用して use security services 設定パラメータを設定します。

use security services が 1 に設定されている場合は、以下の両方の条件が満たされていれば、Adaptive Server はそのセキュリティ・メカニズムをサポートします。

- セキュリティ・メカニズムのグローバル識別子が *interfaces* ファイルまたはディレクトリ・サービス内に登録されている。
- グローバル識別子が、*objectid.dat* 内で、*libtcl.cfg* に登録されているローカル名にマッピングされている。

Adaptive Server が特定のクライアントに使用するセキュリティ・メカニズムを決定する方法については、「クライアントへのセキュリティ・メカニズムの使用」(105 ページ) を参照してください。

統一化ログインの要求

システム・セキュリティ担当者を除いたすべてのユーザに対して、セキュリティ・メカニズムによる認証を行うには、unified login required 設定パラメータを 1 に設定します。次の設定パラメータを設定した場合、sso_role を持つユーザのみがユーザ名とパスワードを使用してサーバにログインできます。

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

たとえば、セキュリティ・メカニズムで認証済みのユーザのログインのみを認めるには、次のコマンドを実行します。

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

セキュア・デフォルト・ログインの確立

セキュリティ・メカニズムからの有効なクレデンシャルを持つユーザが Adaptive Server にログインすると、サーバはそのユーザ名が master..syslogins に存在するかどうかをチェックします。存在する場合、Adaptive Server はそのユーザ名を使用します。たとえば、あるユーザが Kerberos セキュリティ・メカニズムに “ralph” としてログインしたときに、“ralph” という名前が master..syslogins に存在していれば、そのサーバで “ralph” に対して定義されているすべての役割と権限が認められます。

しかし、有効なクレデンシャルを持つユーザであっても、サーバにそのユーザ名が登録されていない場合は、そのユーザが Adaptive Server にログインできるのは sp_configure でセキュア・デフォルト・ログインが定義されている場合だけです。master..syslogins に定義されていないが、セキュリティ・メカニズムによってあらかじめ認証されているユーザには、デフォルト・ログインが使用されます。構文は次のとおりです。

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

secure default login のデフォルト値は “guest” です。

セキュア・デフォルト・ログインは、master..syslogins でも有効なログインでなければなりません。たとえば、“gen_auth” をデフォルト・ログインに設定するには、次の手順に従います。

1　create login を使用して、Adaptive Server での有効なユーザとしてログインを追加します。

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

このプロシージャによって、初期パスワードが “pwgenau” に設定されます。

2　次のように入力して、ログインをセキュア・デフォルトとして指定します。

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

セキュリティ・メカニズムによって認証済みでも Adaptive Server には未登録のユーザには、このログインが使用されます。

注意　このセキュア・デフォルト・ログインに関連付けられている suid は複数のユーザによって使用されます。したがって、デフォルト・ログインによるすべてのアクティビティに対して監査を行うように設定することをおすすめします。また、create loign を使用してすべてのユーザをサーバに登録することも検討してください。

「ログイン・アカウントの作成」(17 ページ) を参照してください。

セキュリティ・メカニズムのログイン名からサーバ名へのマッピング

セキュリティ・メカニズムの中には Adaptive Server で有効でないログイン名を使用できるものもあります。たとえば、30 文字を超えるログイン名や、!、%、*、& などの特殊文字が含まれているログイン名は、Adaptive Server では無効です。Adaptive Server のログイン名は、有効な識別子でなければなりません。『ASE リファレンス・マニュアル』の「第 3 章 式、識別子、およびワイルドカード文字」を参照してください。

表 4-3 は、ログイン名に使用されている無効な文字を Adaptive Server が変換する方法を示します。

表 4-3: ログイン名の無効な文字の変換

無効文字	変換後
アンパサンド & アポストロフィ ’ 円記号 \ コロン : カンマ , 等号 = 左引用符 ‘ パーセント記号 % 右山カッコ > 右引用符 ’ 波型記号 ~	アンダースコア _
脱字記号 ^ 中カッコ { } 感嘆符 ! 左山カッコ < カッコ ( ) ピリオド . 疑問符 ?	ドル記号 $
アスタリスク * マイナス記号 - パイプ | プラス記号 + 引用符 " セミコロン ; スラッシュ / 角カッコ [ ]	シャープ記号 #

暗号化によるメッセージの機密保持の要求

Adaptive Server との間で送受信するすべてのメッセージが暗号化されることを要求するには、msg confidentiality reqd 設定パラメータを 1 に設定します。このパラメータが 0 (デフォルト) の場合、メッセージの機密保持は要求されませんが、機密保持を行うかどうかをクライアント側で設定することは可能です。構文は次のとおりです。

sp_configure *configuration_parameter*, [0 | 1]

たとえば、すべてのメッセージを暗号化するように要求するには、次のコマンドを実行します。

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

データ整合性の要求

msg integrity reqd 設定パラメータを使用して、すべてのメッセージに対して 1 種類以上のデータ整合性チェックを行うことを要求できます。すべてのメッセージについて不正な変更がないかを調べる一般的な検査を行うように要求するには、msg integrity reqd を 1 に設定します。msg integrity reqd が 0 (デフォルト) の場合、メッセージの整合性は要求されませんが、整合性検査がセキュリティ・メカニズムによってサポートされていれば、検査を行うかどうかをクライアント側で設定できます。

ネットワークベース・セキュリティのメモリ要件

1 つのセキュア接続につき約 2K の追加メモリが割り付けられます。max total_memory 設定パラメータの値は、Adaptive Server の起動時に必要とするメモリの量を指定します。たとえば、サーバで 2K の論理ページを使用し、同時に発生するセキュア接続の最大数を 150 と予想する場合は、max total_memory パラメータの値に 150 を追加します。これにより、割り付けられるメモリの量は 2K ブロック 150 個分増加します。

構文は次のとおりです。

sp_configure “max total_memory”, *value*

たとえば、Adaptive Server に必要なメモリが、ネットワークベース・セキュリティ用の追加メモリを含めて 2K ブロック 75,000 個分である場合は、次のコマンドを実行します。

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

『システム管理ガイド 第 2 巻』の「第 3 章 メモリの設定」を参照してください。

統一化ログインをサポートするためのログインの追加

認証済みのクレデンシャルを使用してユーザが Adaptive Server にログインするとき、Adaptive Server は以下の処理を行います。

1　master..syslogins に存在する有効なユーザかどうかをチェックします。そのユーザが master..syslogins に登録されている場合は、Adaptive Server はパスワードを要求しないでログインを承認します。

2　そのユーザ名が master..syslogins に存在しない場合は、デフォルト・セキュア・ログインが定義されているかどうかをチェックします。デフォルト・ログインが定義されていれば、ユーザはデフォルトを使用してログインできます。デフォルト・ログインが定義されていない場合、ユーザはログインできません。

このため、管理者は、有効なログインとして定義されているユーザだけに Adaptive Server の使用を許可するか、ユーザがデフォルト・ログインを使用してログインできるようにするかを決める必要があります。デフォルトを定義するには、デフォルト・ログインを master..syslogins に追加し、sp_configure を使用します。「セキュア・デフォルト・ログインの確立」(99 ページ) を参照してください。

ログインを追加するための一般的な手順

サーバにログインを追加したり、オプションで、ユーザに 1 つ以上のデータベースに対する適切な役割や権限を追加したりするには、表 4-4 に記載されている一般的な手順に従います。

表 4-4: ログインの追加とデータベースへのアクセスの許可

タスク	必要な役割	コマンドまたはプロシージャ	参照箇所
1. ユーザに対応するログインを追加する。	システム・セキュリティ担当者	create login	「ログイン・アカウントの作成」(17 ページ)
2. ユーザを 1 つ以上のデータベースに追加する。	システム管理者またはデータベース所有者	sp_adduser – データベース内からこのプロシージャを実行する。	「データベースへのユーザの追加」(64 ページ)
3. ユーザをデータベースのグループへ追加する	システム管理者またはデータベース所有者	sp_changegroup – データベース内からこのプロシージャを実行する。	• 「ユーザのグループ・メンバシップの変更」(69 ページ) • 『ASE リファレンス・マニュアル』の「sp_changegroup」
4. システム標準の役割をユーザに付与する	システム管理者またはシステム・セキュリティ担当者	grant role	• 「役割の付与と取り消し」(161 ページ) • 『ASE リファレンス・マニュアル』の「grant」
5. ユーザ定義の役割を作成し、作成した役割をユーザに付与する。	システム・セキュリティ担当者	create role grant role	• 「ユーザに対する役割の作成と割り当て」(145 ページ) • 『ASE リファレンス・マニュアル』の「grant」 • 『ASE リファレンス・マニュアル』の「create role」
6. データベース・オブジェクトへのアクセス権を与える。	データベース・オブジェクト所有者		「第 6 章 ユーザ・パーミッションの管理」

リモート接続での Kerberos セキュリティの確立

Adaptive Server は、他のサーバに接続してリモート・プロシージャ・コール (RPC) を実行するときや、コンポーネント統合サービス (CIS) を介したリモート接続で、クライアントとして動作します。

RPC を実行する際の Adaptive Server からのリモート・サーバ・ログインでは、2 つのサーバ間で 1 つの物理接続が確立されます。サーバは、この物理接続を使用して 1 つ以上の「論理接続」、つまり RPC ごとに 1 つの論理接続を確立します。

Adaptive Server は、Kerberos バージョン 5 が提供するクレデンシャル委任機能を使用して CIS からリモート・サーバ接続を試みる Kerberos ログインで、エンドツーエンドの Kerberos 認証をサポートします。

クレデンシャル委任やチケット転送の機能を利用すると、サーバ接続時に Kerberos クライアントによるクレデンシャル委任が可能になり、今後、その他のサーバに接続したときに Kerberos クライアントの代わりにサーバが Kerberos の認証を開始できるようになります。

Adaptive Server に接続されている Kerberos クライアントは、Kerberos クレデンシャル委任機能を使用した CIS を介したリモート Adapter Server への一般的な分散クエリ処理要求で、Adaptive Server のリモート・プロシージャ・コールを要求できます。リモート Adaptive Server への接続に使用される Kerberos 認証機能は、リモート・サーバ・ログインではサポートされません。CIS Kerberos 認証の設定方法の詳細については、『コンポーネント統合サービスユーザーズ・ガイド』の「第 2 章 コンポーネント統合サービスの概要」の「コンポーネント統合サービスのリモート・プロシージャ・コールの設定」を参照してください。

統一化ログインとリモート・サーバ・ログイン

ローカル・サーバとリモート・サーバにセキュリティ・サービスを使用するように設定すると、サーバのセキュリティ・モデルがどちらであっても、以下の 2 つのいずれかの方法で両方のサーバに統一化ログインによるログインが可能です。

- システム・セキュリティ担当者は、リモート・サーバで sp_remoteoption を使用してユーザを “trusted” と定義する。ユーザは「統一化ログイン」を使用してローカル・サーバにアクセスし、リモート・サーバで RPC を実行する。ユーザはリモート・サーバで信頼されている (trusted) ため、パスワードを入力する必要がない。
- ユーザは、ローカル・サーバに接続するときにリモート・サーバのパスワードを指定する。リモート・サーバのパスワードを指定する機能は、ct_remote_pwd routine available with Open Client Client-Library/C にあります。『Open Client Client-Library/C リファレンス・マニュアル』を参照してください。

リモート・サーバ情報の取得

sp_helpserver は、サーバに関する情報を表示します。引数を指定しないで sp_helpserver を実行すると、sysservers に登録されているすべてのサーバについての情報が表示されます。特定のサーバを指定すると、そのサーバに関する情報を表示できます。構文は次のとおりです。

```
sp_helpserver [server]
```

たとえば、GATEWAY サーバに関する情報を表示するには、次のコマンドを実行します。

```
sp_helpserver GATEWAY
```

サーバへの接続とセキュリティ・サービスの使用

isql ユーティリティと bcp ユーティリティでは、以下のコマンドライン・オプションを使用することにより、その接続でネットワークベース・セキュリティ・サービスを有効にすることができます。

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

これらのオプションについて以下で説明します。

- -R *remote_server_principal* は、セキュリティ・メカニズムに対して定義されているサーバのプリンシパル名を指定します。デフォルトでは、サーバのプリンシパル名はサーバのネットワーク名 (-S オプションまたは DSQUERY 環境変数で指定) と一致します。サーバのプリンシパル名とネットワーク名が同じでない場合は、-R オプションを使用する必要があります。
- -V *security_options* は、ネットワークベースのユーザ認証を指定します。このオプションを使用する場合、ユーザはユーティリティを実行する前にネットワークのセキュリティ・システムにログインする必要があります。この場合に、-U オプションを指定するのであれば、セキュリティ・メカニズムに対して定義されているネットワーク・ユーザ名を入力する必要があります。-P オプションで指定したパスワードは無視されます。-V に続く *security_options* 文字列でキー文字オプションを指定することによって、追加のセキュリティ・サービスを有効化することができます。これらのキー文字は、以下のとおりです。
  - c − データ機密性サービスを有効にする。
  - d − クレデンシャル委任を要求し、クライアントのクレデンシャルを転送する。
  - i − データ整合性サービスを有効にする。
  - m − 接続の確立に相互認証を有効にする。

- o – データ・オリジン・スタンピング・サービスを有効にする。
- r – データ・リプレイの検出を有効にする。
- q – 順序不整合の検出を有効にする。

- *-Z security_mechanism* は、接続で使用するセキュリティ・メカニズムの名前を指定します。

セキュリティ・メカニズムの名前は、*libtcl.cfg* 設定ファイルで定義されます。*security_mechanism* の名前が指定されない場合は、デフォルトのメカニズムが使用されます。使用しているプラットフォームの『Open Client/Server 設定ガイド』を参照してください。

Client-Library を使用して Adaptive Server に接続する場合は、サーバに接続する前にセキュリティ・プロパティを定義できます。たとえば、メッセージの順序をチェックするには、CS_SEC_DETECTSEQ プロパティを設定します。セキュリティ・サービスを Client-Library とともに使用する方法については、『Open Client Client-Library/C リファレンス・マニュアル』を参照してください。

クライアントへのセキュリティ・メカニズムの使用

Adaptive Server は、起動時に、サポートするセキュリティ・メカニズムを決定します「サポートされているセキュリティ・サービスとメカニズムに関する情報」(106 ページ) を参照してください。Adaptive Server は、サポートするセキュリティ・メカニズムのリストから、特定のクライアントに使用するセキュリティ・メカニズムを選択する必要があります。

クライアントがセキュリティ・メカニズムを指定した場合 (たとえば isql の -Z オプション)、Adaptive Server はそのセキュリティ・メカニズムを使用します。クライアントによる指定がない場合は、*libtcl.cfg* ファイルにリストされている最初のセキュリティ・メカニズムを使用します。

使用できるセキュリティ・サービスの情報の取得

Adaptive Server では、次の情報を取得できます。

- Adaptive Server がサポートしているセキュリティ・メカニズムとセキュリティ・サービス
- 現在のセッションに対してアクティブなセキュリティ・サービス
- 特定のセキュリティ・サービスがセッションに対して有効にされているかどうか

サポートされているセキュリティ・サービスとメカニズムに関する情報

システム・テーブル syssecmechs には、Adaptive Server がサポートしているセキュリティ・メカニズムとセキュリティ・サービスについての情報が格納されています。これは、検索の実行時に動的に作成されるテーブルで、以下のカラムがあります。

- sec_mech_name − セキュリティ・メカニズムの名前。“NT LANMANAGER” など。
- available_service − セキュリティ・メカニズムがサポートするセキュリティ・サービスの名前。“unified login” など。

このテーブルでは、1 つのセキュリティ・メカニズムに複数のローが存在することがあり、各ローはそのメカニズムでサポートされている個々のセキュリティ・サービスを示します。

Adaptive Server がサポートしているすべてのセキュリティ・メカニズムとセキュリティ・サービスのリストを表示するには、次のクエリを実行します。

```
select * from syssecmechs
```

アクティブなセキュリティ・サービスに関する情報

現在のセッションでどのセキュリティ・サービスがアクティブかを調べるには、show_sec_services という関数を使用します。

```
select show_sec_services()

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

有効なセキュリティ・サービスに関する情報

“mutualauth” などの特定のセキュリティ・サービスを有効にするかどうかを決定するには、is_sec_service_on 関数を使用します。ここで、*security_service_nm* は有効になるセキュリティ・サービスです。

is_sec_service_on(*security_service_nm*)

syssecmechs を問い合わせたときに返されるセキュリティ・サーバを使用します。

たとえば、“mutualauth” (相互認証) が有効かどうかを調べるには、次のコマンドを実行します。

```
select is_sec_service_on("mutualauth")

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

結果が 1 の場合は、このセッションではこのセキュリティ・サービスが有効です。結果が 0 の場合は、このセキュリティ・サービスは使用されていません。

Kerberos の使用

Kerberos は、シークレット・キー暗号法を使用するネットワーク認証プロトコルであり、これによってクライアントがネットワーク接続経由でサーバに ID を証明できます。ユーザがオペレーティング・システムにログインしたとき、または認証プログラムを実行することにより、ユーザ・クレデンシャルが取得されます。このクレデンシャルは、認証を実行するときに各アプリケーションによって使用されます。ユーザは 1 回ログインすれば各アプリケーションにログインする必要はありません。

Kerberos は、KDC (Key Distribution Center) が稼動しており、レルムに対して適切に設定されていることと、クライアント・ライブラリがレルム内の各クライアント・ホストにインストールされていることを前提としています。設定の詳細については、Kerberos のマニュアルと Kerberos ソフトウェアに付属するリファレンス・ページを参照してください。

Adaptive Server では、Kerberos は次のようにサポートされます。

- CyberSafe Kerberos ライブラリ
- MIT Kerberos ライブラリ・バージョン 1.3.1
- ネイティブ・ライブラリ

注意 Kerberos セキュリティ・オプションを有効にするには、“Security and directory services” パッケージである ASE_SECDIR が必要です。

Kerberos の互換性

表 4-5 は、各種 Kerberos がサポートされるプラットフォームを示します。

表 4-5: Adaptive Server における Kerberos の相互運用性

ハードウェア・プラットフォーム	KDC サーバ	GSS (Generic Security Standard) クライアント
Solaris 32	CSF、AD、MIT	CSF、MIT、ネイティブ
Solaris 64	CSF、AD、MIT	CSF、MIT、ネイティブ
Linux 32	CSF、AD、MIT	MIT、ネイティブ
Windows 32	CSF、AD	CSF
AIX 32	CSF	CSF

この相互運用性の表では次の略称を使用しています。

- CSF － CyberSafe 社
- AD － Microsoft Active Directory
- MIT － MIT バージョン 1.3.1

Kerberos 環境での Adaptive Server の起動

Kerberos 環境で Adaptive Server を起動するには、Adaptive Server 名を KDC に追加して、サービス・キーをキー・テーブル・ファイルに抽出します。次に例を示します。

```
/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.
```

注意 管理者は、コマンド・ラインでパスワードを指定する方法で認証を受けることもできます。この例では **-k** オプションを使用しています。これは、パスワード入力のプロンプトを表示するのではなく、**-t** オプションで指定した */krb5/v5srvtab* ファイルの中で管理者と Adaptive Server のキーを検索することを管理者に指示するものです。この方法は、シェル・スクリプトを作成する場合に便利です。

Kerberos の設定

設定プロセスは、使用する Kerberos の種類に関係なく共通です。

1 サードパーティ製 Kerberos ソフトウェアを設定して、Kerberos 管理ユーザを作成します。これには、次の処理を行います。

   a Kerberos クライアント・ソフトウェアを、Open Client Server クライアントまたは Adaptive Server が稼働するマシンにインストールします。次のクライアント・パッケージは動作が確認されています。

      - CyberSafe TrustBroker 4.0
      - MIT Kerberos バージョン 1.3.1

   b Kerberos KDC サーバを別の専用マシンにインストールします。

注意 CyberSafe TrustBroker 4.0、MIT Kerberos v.1.3.1、Microsoft Windows Active Directory の KDC は、Adaptive Server とともに使用できることが確認されています。

c　Kerberos サーバに、管理権限を持つ管理者アカウントを作成します。このアカウントは、後のクライアント作業 (クライアント・マシンでのプリンシパルの作成など) で使用します。

注意　この後の手順は Kerberos クライアント・マシンで実行します。

2　Adaptive Server の Kerberos プリンシパル *ase120srv* または *ase120srv@MYREALM* を追加します。

3　プリンシパル *ase120srv@MYREALM* の *keytab* ファイルを抽出し、次のようにファイルとして保存します。

```
/krb5/v5srvtab
```

次の UNIX の例では、CyberSafe または MIT Kerberos で利用可能なコマンド・ライン・ツール kadmin を使用します。Kerberos とユーザを管理する GUI ツールもあります。

```
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.
```

運用環境では、*keytab* ファイルへのアクセスを制御してください。*keytab* ファイルの読み込みを許可されているユーザは、使用しているサーバになり代わるサーバを作成できます。

chmod と chgrp を使用して、*/krb5/v5srvtab* を次のように設定します。

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

Active Directory を KDC として使用するときは、Domain Controller にログインしてユーザと Adaptive Server プリンシパルを追加します。Active Directory ユーザとコンピュータ・ウィザードを使用して、ユーザとプリンシパルを作成できます。

Adaptive Server で使用する *keytab* ファイルを抽出するには、ktpass というオプション・ツールが必要です。これは、Microsoft サポート ツール・パッケージに含まれています。

Active Directory を使用する場合、ktpass による *keytab* の抽出は、プリンシパルの作成とは別に実行します。Adaptive Server の *keytab* ファイルは、Windows では CyberSafe プログラム・ファイルと同じ場所にあります。たとえば、CyberSafe ソフトウェアが C ドライブにインストールされている場合、Adaptive Server の *keytab* ファイルは *c:\Program Files\CyberSafe\v5srvtab* に格納されると考えられます。

4 ユーザ “sybuser1” の Kerberos プリンシパルを “sybuser1@MYREALM” として追加します。

5 Adaptive Server を起動し、isql を使用して “sa” としてログインします。この後の手順で、Kerberos セキュリティ・サービスを使用するための Adaptive Server パラメータを設定し、ユーザのログイン・アカウントを作成します。この手順は Windows マシンでも UNIX マシンでも同じです。

- 設定パラメータ use security services を 1 に変更します。

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

- ユーザ “sybuser1” のために新しいログインを追加してから、ユーザを追加します。

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

6 Adaptive Server を停止し、管理ファイルと接続設定ファイルを変更します。

- UNIX プラットフォームの場合 － *$SYBASE/* 内に *interfaces* ファイルがあり、次のようなエントリが含まれています。

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

  Windows プラットフォームの場合 － *%SYBASE%\ini* 内に *sql.ini* ファイルがあり、次のように同様のサーバ・エントリが含まれています。

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

- UNIX プラットフォームでは、*$SYBASE/$SYBASE_OCS/config/* に *libtcl.cfg* ファイルまたは *libtcl64.cfg* ファイルがあります。SECURITY セクションに、CyberSafe Kerberos クライアント・ライブラリに関する次のようなエントリが含まれます。

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

64 ビット版 CyberSafe Kerberos クライアント・ライブラリのエントリは次のようになります。

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

MIT Kerberos クライアント・ライブラリを使用するマシンでは、エントリは次のようになります。

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

OS 提供のネイティブ・ライブラリを使用するマシン (Linux など) では、エントリは次のようになります。

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

Windows の場合 － *%SYBASE%\%SYBASE_OCS%\ini\libtcl.cfg* ファイルに次のようなエントリが含まれます。

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

注意　使用する GSS API ライブラリは、`libgss=<gss shared object path>` によって指定されます。複数のバージョンの Kerberos Client ライブラリが 1 台のマシンにインストールされている場合は特に、使用するライブラリのロケーションを明確に指定する必要があります。

- また、*$SYBASE/$SYBASE_OCS/config/* の *objectid.dat* を調べて、*[secmech]* セクションに *csfkrb5* のエントリがあることを確認します。

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

7　環境変数を使用して、*keytab* ファイル、Kerberos 設定ファイル、レルム設定ファイルのデフォルト・ロケーションを無効にできます。これは Kerberos 固有の動作であり、すべてのプラットフォームで同様に機能するとは限りません。

たとえば、CyberSafe UNIX プラットフォームでは、CSFC5KTNAME 環境変数を使用して *keytab* ファイルを指定します。

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

MIT Kerberos でこれに相当する環境変数は KRB5_KTNAME です。

これらの環境変数の詳細については、各ベンダのマニュアルを参照してください。

場合によっては、ダイナミック・ライブラリ検索パスの環境変数を変更する必要があります。UNIX で一般的に使用される環境変数は LD_LIBRARY_PATH です。Windows では通常、PATH が DLL のロケーションを指すように設定されています。アプリケーションでサードパーティのオブジェクトを正しくロードするには、これらの環境変数を変更する必要があります。たとえば、次のコマンドを使用すると、CyberSafe 32 ビット版の *libgss.so* 共有オブジェクトのロケーションが C シェル環境の検索パスに追加されます。

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

8 Adaptive Server を再起動します。次のメッセージが表示されます。

```
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 isql を使用して UNIX ユーザ “sybuser1” として次のように接続します (-U 引数と -P 引数は使用しません)。

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

次のように暗号化オプションを使用することもできます。

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

プリンシパル名の使用

プリンシパル名は、Kerberos KDC (Key Distribution Center) で認証するときにサーバが使用する名前です。複数の Adaptive Server インスタンスを実行中の場合は、Adaptive Server ごとに異なるプリンシパル名を使用する必要があります。

Adaptive Server プリンシパル名の指定

Adaptive Server の名前を指定するには、環境変数 DSLISTEN と DSQUERY、またはコマンドライン・オプション dataserver -s*server_name* を使用します。

プリンシパル名を設定するには、setenv コマンドまたは -k dataserver オプションを使用します。

デフォルトのプリンシパル名は Adaptive Server の名前です。別の名前を指定するには、Adaptive Server を起動して Kerberos を使用する前に、SYBASE_PRINCIPAL を次のように設定します。

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

Adaptive Server のプリンシパル名を設定すると、Adaptive Server はこの変数の値を使用して自身を Kerberos で認証します。

Adaptive Server の起動時に Adaptive Server のプリンシパル名を指定するには、次のコマンドを使用します。

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

Adaptive Server を Kerberos セキュリティ・メカニズムを有効にして起動する場合、Adaptive Server では最初に Kerberos 認証の -k オプションで指定されているプリンシパル名が使用されます。-k オプションが指定されていない場合、Adaptive Server は環境変数 SYBASE_PRINCIPAL でプリンシパル名を確認します。どちらも指定されていない場合、Adaptive Server は認証にサーバ名を使用します。

Adaptive Server では、プリンシパル名のエントリが *keytab* ファイル内に存在する場合に、別のサーバのプリンシパル名を使用する Kerberos Open Client 接続を使用できます。別のプリンシパル名による接続を許可するには、次のいずれかを行います。

- -k オプションのパラメータとして空の文字列を渡す。
- 環境変数 SYBASE_PRINCIPAL を "" に設定する。次に例を示す。

```
export SYBASE_PRINCIPAL=""
```

例

この例では、Adaptive Server の名前が “secure_ase”、レルム名は “MYREALM.COM” であり、Adaptive Server の名前は、-s パラメータを使用したコマンド・ラインで dataserver に指定されます。現在のレルムは、secbase 属性値により *libtcl.cfg* で指定されます。

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

Adaptive Server の *keytab* ファイルで定義されたプリンシパル名が “aseprincipal@MYREALM.COM” の場合、次のオプション 1 または 2 を使用してサーバのプリンシパル名を設定し、デフォルトの Adaptive Server プリンシパル名を上書きできます。

- オプション 1 － -k " を指定する。

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

  Kerberos での認証に使用される Adaptive Server のプリンシパル名は “aseprincipal@MYREALM.COM” です。

- オプション 2 － SYBASE_PRINCIPAL を設定する。

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

Kerberos での認証に使用される Adaptive Server のプリンシパル名は、*$SYBASE_PRINCIPAL* の値の “aseprincipal@MYREALM.COM” です。

- オプション 3 － -k と SYBASE_PRINCIPAL のいずれも設定しない。

```
% $SYBASE/$SYBASE_ASE/bin/dataserver -dmaster.dat
-s secure_ase
```

Kerberos での認証に使用される Adaptive Server のプリンシパル名は “secure_ase@MYREALM.COM” です。

sybmapname を使用したユーザ・プリンシパル名の処理

sybmapname は、Kerberos 環境で使用される外部のユーザ・プリンシパル名を Adaptive Server のユーザ・ログインのネームスペースに変換します。sybmapname 共有オブジェクトをカスタマイズして、Kerberos 入力バッファで指定された名前を、Adaptive Server 出力バッファへのログインに適した名前にマップできます。

ユーザのプリンシパル名と Adaptive Server のログイン名との間でカスタム・マッピングを実行するには、sybmapname 共有オブジェクトを使用します。この共有オブジェクトは、オプションでサーバの起動時にロードされ、共有オブジェクトに含まれている関数 syb__map_name は、Kerberos 認証が成功した後およびユーザ・プリンシパルが syslogins テーブル内のログインにマップされる直前に呼び出されます。この関数は、マップされるユーザのプリンシパル名とログイン名が同一ではない場合に役に立ちます。

```
syb__map_name(NAMEMAPTYPE *protocol, char *orig,
int origlen, char *mapped, int *mappedlen)
```

各パラメータの意味は、次のとおりです。

- NAMEMAPTYPE *protocol － この関数の使用のために予約されている構造体を表す。
- char *orig － Null で終了しない入力バッファ。
- int origlen － 入力バッファの長さ。255 文字以内にする必要がある。
- char *mapped － Null で終了しない出力バッファ。
- int *mappedlen － 出力バッファの長さ。30 文字以内にする必要がある。

syb__map_name は、マッピングが成功した場合は 0 よりも大きい値を返し、マッピングが実行されなかった場合は 0 の値を返し、syb__map_name でエラーが発生した場合は 0 よりも小さい値を返します。エラーが発生すると、Adaptive Server のエラー・ログにマッピングの失敗をレポートするメッセージが書き込まれます。

たとえば、Adaptive Server で Kerberos ユーザを認証するには、次の手順に従います。

1　Kerberos セキュリティ・メカニズムを使用するように Adaptive Server を設定します。「Kerberos の使用」(107 ページ) と Open Client/Server マニュアル、および Sybase Web サイト (http://www.sybase.com/detail?id=1029260)のホワイト・ペーパー「Configuring Kerberos for Sybase」を参照してください。

サンプルの *sybmapname.c* ファイルは、*$SYBASE/$SYBASE_ASE/sample/server/sybmapname.c* にあります。

2　*sybmapname.c* を修正して、ロジックを実装します。「sybmapname を使用する場合の注意事項」(117 ページ) を参照してください。

3　提供されている汎用プラットフォーム固有の makefile を使用して共有オブジェクトまたは DLL を構築します。makefile は、プラットフォーム固有の設定に合わせて変更しなければならない場合があります。

4　生成された共有オブジェクトは、UNIX マシンでは $LD_LIBRARY_PATH で指定したロケーション、Windows マシンでは PATH 変数で指定したロケーションに保存されます。ファイルには、“sybase” ユーザに対する読み取りおよび実行パーミッションが必要です。

注意　“sybase” ユーザにのみ読み取りパーミッションや実行パーミッションを許可し、他のアクセスはすべて拒否することをおすすめします。

Kerberos 認証を使用した Adaptive Server へのログインの確認

Kerberos 認証を使用して Adaptive Server へのログインを確認するには、次のことを前提とします。

- *$SYBASE* は、リリースおよびインストール・ディレクトリを参照する。
- *$SYBASE_ASE* は、サーバ・バイナリを含む Adaptive Server バージョン・ディレクトリを参照する。
- *$SYBASE_OCS* は、Open Client/Server バージョン・ディレクトリを参照する。

例 1　クライアントのプリンシパル名が `user@REALM` であり、syslogins テーブル内の対応するエントリが `user_REALM` である場合は、入力文字列 `user@realm` を受け取って、その入力文字列を出力文字列 `user_REALM` に変換するように sybmapname をコード化できます。

例 2　クライアントのプリンシパル名が `user` の場合、および syslogins テーブルの対応するエントリが `USER` の場合、入力文字列 `user` を受け取って、この文字列を大文字の文字列 `USER` に変換するように sybmapname をコード化できます。

sybmapname は、Adaptive Server によって実行時に読み込まれ、そのロジックを使用して必要なマッピングを実行します。

次の操作と出力は、例 2 で説明する sybmapname 関数を示しています。syb__map_name() に対してカスタマイズされた定義を含む *sybmapname.c* ファイルはコンパイルして、共有オブジェクト (または DLL) としてビルドした後に、適切なパスのロケーションに保存する必要があります。Kerberos のセキュリティ・メカニズムを有効にして Adaptive Server を起動します。

TGT (Ticket Granted Ticket) は、識別情報を提供する、暗号化形式のファイルです。このファイルを初期化するには、次のように入力します。

```
$ /krb5/bin/kinit johnd@public
Password for johnd@public:
$
```

TGT を一覧表示するには、次のように入力します。

```
$ /krb5/bin/klist
   Cache Type: Kerberos V5 credentials cache
   Cache Name: /krb5/tmp/cc/krb5cc_9781
Default principal: johnd@public
```

“sa” としてログインし、“johnd” のユーザ・ログインを確認します。

```
$ $SYBASE/$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 認証が成功すると、sybmapname ユーティリティを使用して小文字の johnd が大文字の JOHND にマップされ、ユーザ johnd は Adaptive Server にログインできるようになります。

```
$ $SYBASE/$SYBASE_OCS/bin/isql -V -I'pwd'/interfaces
1>
```

sybmapname を使用する場合の注意事項

sybmapname のコーディングを行う場合は、次の点に注意する必要があります。

- サンプルの *sybmapname.c* プログラムに変更を加える場合は、慎重に行う必要があります。セグメンテーション・フォールトを発生させるコード、exit を呼び出すコード、system calls を呼び出すコード、UNIX シグナルを変更するコード、ブロック呼び出しを行うコードの使用は避けてください。不適切なコーディングや呼び出しは、Adaptive Server エンジンを妨害する場合があります。

  注意 sybmapname におけるコード・エラーは、Sybase の責任ではありません。

- コードを注意深く作成し、すべてのポインタをチェックしてから参照を解除して、システム・コールを回避します。記述する関数は、クイック・ネーム・フィルタリング関数にする必要があります。
- goto 文を使用しないでください。プラットフォームによっては、これらの文によって予期しない悪影響を受ける場合があります。
- 複数のレルムを使用する場合は、ユーザ・プリンシパル名を適切なログイン名に注意深くマップし、レルム情報が反映されるようにします。たとえば、ユーザ・プリンシパル名 `userA@REALMONE` と `userB@REALMTWO` をそれぞれ持つ 2 人のユーザがいる場合、ログイン名 `userA_REALMONE` と `userB_REALMTWO` にマップします。`userA` または `userB` にはマップしないでください。この動作により、異なるレルムに属する 2 人のユーザが区別されます。

Kerberos による同時認証

以前のバージョンでは、Kerberos による認証時にロック・メカニズムを使用することによって内部データ構造を保護していましたが、Adaptive Server バージョン 15.0.3 では、Kerberos による同時認証がサポートされるようになりました。

Kerberos 認証を使用した同時ログインがある場合は、Adaptive Server によって複数の Kerberos 認証セッションが確立されます。

バージョン 15.0.3 では、Kerberos による認証時に同時ログイン・セッションがブロックされる問題も解決されています。同時実行性に関連したこの問題は、以前のバージョンの Adaptive Server を、MIT バージョン 1.3.x および 1.4.x の Kerberos GSSAPI ライブラリとともに使用する場合に発生します。

LDAP ユーザ認証のための Adaptive Server の設定

LDAP ユーザ認証を使用すると、クライアント・アプリケーションは Adaptive Server にユーザ名とパスワードの情報を送信し、syslogins ではなく LDAP サーバによる認証を行えるようになります。LDAP サーバを使用する認証では、Adaptive Server またはアプリケーション固有のパスワードではなく、サーバ全体のパスワードを使用できます。

LDAP ユーザ認証は、ユーザ管理を単純化して集中化する場合や、ユーザ管理が煩雑にならないようにする場合に最適な方法です。

LDAP ユーザ認証は、LDAP プロトコル標準バージョン 3 に準拠したディレクトリ・サーバ (Active Directory、iPlanet、OpenLDAP Directory Server など) で動作します。

LDAP ユーザ認証では、次のいずれかの認証アルゴリズムを使用します。

- 生成 DN (認証用、Adaptive Server バージョン 12.5.1 以降で使用可能)
- 検索 DN (Adaptive Server バージョン 12.5.2 以降で使用可能)

各アルゴリズムは、ユーザの DN (識別名) を取得する方法が異なります。

LDAP プロトコルで使用されるプライマリ・データ構造は LDAP URL です。

LDAP URL は、LDAP サーバ上のオブジェクトまたは値のセットを指定します。Adaptive Server は、LDAP URL を使用して、ログイン要求の認証に使用する LDAP サーバと検索基準を指定します。

LDAP URL では、次の構文を使用します。

```
ldapurl::=ldap://host:port/node/attributes [base | one | sub] filter
```

各要素の意味は次のとおりです。

- *host* − LDAP サーバのホスト名。
- *port* − LDAP サーバのポート番号。
- *node* − 検索を開始するオブジェクト階層内でのノードを指定する。
- *attributes*− 結果セットで返す属性のリスト。属性リストは、LDAP サーバによって異なることがある。
- base | one | sub − 検索条件を修飾する。base は、ベース・ノードの検索を指定する。one は、node で指定されたベース・ノードとその 1 つ下のレベルのノードの検索を指定する。sub は、node で指定されたベース・ノードとその下位レベルのすべてのノードの検索を指定する。
- filter − 認証する属性を指定する。フィルタは、`uid=*` のように簡潔にしたり、`(uid=*)(ou=group)` のように複雑にすることもできる。

生成 DN アルゴリズム

生成 DN アルゴリズムを使用する場合、ログインは次の手順で行われます。

1　Open Client は、Adaptive Server のリスナ・ポートに接続します。

2　Adaptive Server リスナは、接続を受け付けます。

3　Open Client は、内部ログイン・レコードを送信します。

4　Adaptive Server は、ログイン・レコードを読み込みます。

5　Adaptive Server は、プライマリ URL から生成した DN とログイン・レコードのログイン名を使用して LDAP サーバにバインドします。このとき、ログイン・レコードのパスワードも使用します。

6　LDAP サーバは、ユーザを認証し、成功か失敗かを示すメッセージを返します。

7　プライマリ URL で検索が指定されている場合、Adaptive Server は LDAP サーバに検索要求を送信します。

8　LDAP サーバは、検索結果を返します。

9　Adaptive Server は、検索結果に基づいてログインを受け付けるか、または拒否します。

検索 DN アルゴリズム

検索 DN アルゴリズムを使用する場合、ログインは次の手順で行われます。

1　Open Client は、Adaptive Server のリスナ・ポートに接続します。

2　Adaptive Server リスナは、接続を受け付けます。

3　Open Client は、内部ログイン・レコードを送信します。

4　Adaptive Server は、ログイン・レコードを読み込みます。

5　Adaptive Server は、ディレクトリ・サーバのアクセス・アカウントを使用して LDAP サーバにバインドします。

手順 5 ～ 6 で確立された接続は、次に Adaptive Server が認証を試行して DN 検索への接続を再利用するまで継続します。

6　LDAP サーバは、ユーザを認証し、成功か失敗かを示すメッセージを返します。

7　Adaptive Server は、ログイン・レコードのログイン名と DN 検索 URL に基づいて、LDAP サーバに検索要求を送信します。

8　LDAP サーバは、検索結果を返します。

9　Adaptive Server は、検索結果を読み込み、DN 検索 URL から属性値を取得します。

10 Adaptive Server は、取得した属性値を DN として使用し、パスワードを使用して LDAP サーバにバインドします。

11 LDAP サーバは、ユーザを認証し、成功か失敗かを示すメッセージを返します。

12 プライマリ URL で検索が指定されている場合、Adaptive Server は LDAP サーバに検索要求を送信します。

13 LDAP サーバは、検索結果を返します。

14 Adaptive Server は、検索結果に基づいてログインを受け付けるか、または拒否します。

上記のいずれかの認証基準が満たされない場合、Adaptive Server は一般的なログインの失敗をレポートします。

プライマリ URL 文字列またはセカンダリ URL 文字列の検索基準を指定しない場合は、手順 12 ～ 13 を省略できます。認証が完了し、手順 11 で返される成功か失敗かを示すメッセージが表示されます。

LDAP の設定

Adaptive Server を LDAP 認証向けに設定し、既存の Adaptive Server を LDAP にマイグレートすることができます。

❖ 新しい Adaptive Server での LDAP の設定

1 Adaptive Server の LDAP URL 検索文字列とアクセス・アカウントの値を指定します。

2 enable ldap user auth を 2 に設定します。

3 LDAP ベンダ提供のツールを使用して、LDAP ディレクトリ・サーバにユーザを追加します。

4 create login を使用して Adaptive Server にユーザを追加します。また、sp_maplogin を使用すると、認証時にログイン・アカウントが自動的に作成されるように設定したり、他のログイン制御を適用したりできます。

❖ 既存の Adaptive Server の LDAP へのマイグレーション

既存のサーバでサービスが中断されないようにするには、次の操作を実行して Adaptive Server を LDAP にマイグレートします。

1 Adaptive Server に LDAP URL 検索文字列を指定します。

2 設定パラメータ enable ldap user auth を 1 に設定します。

3 LDAP ディレクトリ・サーバにユーザを追加します。

4　すべてのユーザを LDAP サーバに追加するときに、すべての認証が LDAP で行われるようにするには、enable ldap user auth を 2 に設定するか、sp_maplogin を使用してログイン制御で設定パラメータを上書きします。

LDAP ユーザ認証の管理

LDAP URL 検索文字列の作成または表示、LDAP URL 検索文字列またはログインの確認、アクセス・アカウントとチューニング可能な LDAPUA (LDAP ユーザ認証) 関連パラメータの指定には、sp_ldapadmin を使用します。sp_ldapadmin を実行するには、システム・セキュリティ担当者 (SSO) の役割が必要です。

詳細については、『ASE リファレンス・マニュアル：コマンド』を参照してください。

生成 DN アルゴリズムの例

使用する LDAP サーバのトポロジとスキーマが単純な場合は、ユーザ認証に生成 DN アルゴリズムを使用できます。商用のスキーマ (iPlanet ディレクトリ・サーバや OpenLDAP ディレクトリ・サーバなど) を使用する場合、ユーザは LDAP サーバ・ツリー内の同じコンテナ内のオブジェクトとして作成され、このオブジェクトのロケーションに基づいてユーザの DN が決定されます。ただし、LDAP サーバのスキーマには以下の制限があります。

- 認証されるユーザをユニークに識別する属性名を含むフィルタを指定する必要がある。
- 属性 `name=*` を含むフィルタを指定する必要がある。アスタリスクはワイルドカード文字。フィルタに使用する属性名は、LDAP サーバのスキーマによって異なる。
- Adaptive Server のログイン名は、UNIX ユーザ名などと同様の短縮ユーザ名である。
- DN は、埋め込みスペースや句読表記を含むフル・ネームではなく、短縮ユーザ名を使用する。たとえば、jqpublic は DN の制限事項を満たしているが、“John Q. Public” は満たしていない。

iPlanet の例

LDAP のベンダによっては、以下の例で使用している以外のオブジェクト名、スキーマ、属性を使用することがあります。使用できる LDAP URL 検索文字列は数多くあります。また、有効なサイトがスキーマをローカルに拡張したり、サイトごとに異なる方法でスキーマを使用したりすることもできます。

- 次の例では、`uid=*` フィルタを使用しています。Adaptive Server は、このフィルタのワイルドカードを認証対象となる Adaptive Server のログイン名に置換してから、LDAP URL のノード・パラメータに追加して DN を生成します。生成される DN は次のとおりです。

```
uid=myloginname,ou=People,dc=mycomany,dc=com
```

- Adaptive Server は、バインド操作に成功した後、接続を使用して uid などの属性名を検索します。この属性名は、ログイン名と同じです。

```
sp_ldapadmin set_primary_url,
'ldap://myhost:389/ou=People,dc=mycompany,dc=com??sub?uid=*'
```

- 次の例では、OpenLDAP 2.0.25 で定義された、属性名 cn を含むスキーマを使用しています。

  生成 DN は cn=myloginname,dc=mycompany,dc=com です。

```
sp_ldapadmin set_primary_url,
'ldap://myhost:389/dc=mycompany,dc=com??sub?cn=*'
```

検索 DN アルゴリズムの例

生成 DN アルゴリズムを使用するための制限事項を満たしていない Active Directory サーバまたはその他の LDAP サーバ環境を使用する場合は、検索 DN アルゴリズムを使用します。

- Windows 2000 Server で提供されている商用のユーザ・スキーマを使用する Active Directory サーバの場合は、以下の手順を実行します。

  a アクセス・アカウント情報を設定します。

```
sp_ldapadmin set_access_acct,
'cn=Admin Account, cn=Users, dc=mycompany, dc=com',
'Admin Account secret password'
```

  b プライマリ URL を設定します。

```
sp_ldapadmin set_primary_url, 'ldap://hostname:389/
```

  c DN 検索 URL 検索文字列を設定します。

```
sp_ldapadmin set_dn_lookup_url,
'ldap://hostname:389/cn=Users,dc=mycompany,dc=com?distinguishedName?one?samaccountname=*'
```

Windows 2000 では、通常、短縮名は「ユーザ・ログオン名」と呼ばれ、デフォルト・スキーマで属性名 samaccountname を割り当てられています。これは、Adaptive Server のログイン名と一致させるために使用する属性名です。この属性名を使用して、Adaptive Server のログイン名が検索されます。ユーザの DN には、句読表記と埋め込みスペースを含むフル・ネーム (たとえば、cn=John Q. Public, cn=Users, dc=mycomany, dc=com) が使用されます。Windows の DN では短縮名を使用しないため、検索 DN アルゴリズムは、LDAP サーバに Active Directory スキーマ (デフォルト) を使用しているサイトに適しています。プライマリ URL は検索を指定しません。代わりに、バインド操作を使用して認証を行います。

検索フィルタによる Adaptive Server へのアクセスの制限例

LDAP URL 検索文字列を使用して、LDAP サーバ上の特定のユーザ・グループだけにアクセスを制限できます。たとえば、accounting グループのユーザだけがログインできるようにするには、複合フィルタを使用して、属性が group=accounting のユーザのグループだけにアクセスを制限します。

- 次の LDAP URL 文字列では、iPlanet サーバに生成 DN アルゴリズムを使用しています。

```
sp_ldapadmin set_primary_url,
'ldap://myhost:389/ou=People,dc=mycompany,
dc=com??sub?(&(uid=*)(group=accounting))'
```

Adaptive Server は、`uid=mylogin,ou=People,dc=mycompany,dc=com` という DN を使用してバインドします。この ID を使用したバインドが成功すると、Adaptive Server は次のように検索します。

```
"ou=People,dc=mycompany,dc=com??sub?(&(uid=mylogin)(group=accounting))"
```

この検索からオブジェクトが返されると、認証が成功します。

- 以下の例では、LDAP URL 検索文字列と複合フィルタを使用しています。

```
sp_ldapadmin set_primary_url,
'ldap://myhost:389/ou=people,dc=mycompany,dc=com??sub?(&(
uid=*)(ou=accounting) (l=Santa Clara))'

sp_ldapadmin, set_primary_url,
'ldap://myhost:389/ou=people,dc=mycompany,dc=com??sub?(&(
uid=*)(ou=Human%20Resources))'
```

LDAP ユーザ認証パスワード情報の変更

Adaptive Server が LDAP サーバから取得してクライアントに渡す、LDAP ユーザ認証関連の通知メッセージが 2 つあります。

- 期限が切れそうな LDAP ユーザ認証パスワードを使用する LDAP 認証メカニズムを使用して Adaptive Server にログインした場合は、次のメッセージが表示される。

  パスワードはあと *<number>* 日で有効期限が切れます。

- LDAP サーバ管理者がパスワードをリセットした後、または LDAP サーバのパスワードの期限が切れた後に、LDAP 認証メカニズムを使用して Adaptive Server にログインすると、次のメッセージ 4002 が表示される。

  ログインに失敗しました

  次のように、監査が有効で、errors 監査オプションがオンになっている場合は、メッセージ 4099 が監査ログに送信される。

  LDAP パスワードの期限が切れました。

注意　この追加の情報を提供できるように LDAP サーバを設定してください。また、Adaptive Server は、LDAP クライアントに対する LDAP パスワード制御の転送をサポートしている必要があります。

フェールオーバのサポート

プライマリ URL で指定された LDAP ディレクトリ・サーバで重大な障害が発生し、ネットワーク要求に応答しなくなった場合、Adaptive Server はセカンダリ URL で指定されたセカンダリ LDAP ディレクトリ・サーバに接続しようとします。Adaptive Server は、LDAP 関数 ldap_init を使用して、LDAP ディレクトリ・サーバへの接続をオープンできるかどうかを調べます。プライマリ URL 文字列が NULL または無効である場合、Adaptive Server はセカンダリ URL へのフェールオーバを試行します。LDAP のバインド操作や検索操作で障害が発生した場合、Adaptive Server はセカンダリ URL にフェールオーバしません。

Adaptive Server ログインと LDAP ユーザ・アカウント

LDAP ユーザ認証を有効にし、認証アルゴリズムと URL 文字列の選択と設定を行ったら、ユーザ・アカウントを設定します。LDAP 管理者が LDAP サーバのアカウントの作成と管理を行い、データベース管理者が Adaptive Server のアカウントの作成と管理を行います。また、データベース管理者は、管理オプションを使用して、Adaptive Server と LDAP サーバなどの外部認証メカニズムを統合するときのログイン・アカウントを柔軟に設定できます。データベース管理者は、従来のコマンドとプロシージャを使用して、Adaptive Server アカウントの役割、デフォルト・データベース、デフォルト言語、およびその他のログイン固有の属性の管理を続行できます。

表 4-6 は、ログイン時の syslogins テーブルの変更を示します。ここに示す変更は、LDAP ユーザ認証が設定済みで、ログインが LDAP の使用を制限されておらず、create login マッピングを設定していないことを前提としています。

表 4-6: LDAP による syslogins の変更

syslogins にそのユーザのローが既に存在する	LDAP サーバ認証に成功	syslogins の変更
いいえ	はい	変更なし、ログインは失敗
いいえ	いいえ	変更なし、ログインは失敗
はい	はい	パスワードが変更された場合は、ローが更新される
はい	いいえ	変更なし

セカンダリ検索サーバのサポート

Adaptive Server では、LDAP サーバによって認証された Adaptive Server クライアントの継続的なサポートが提供されます。LDAP サーバで障害が発生した場合や、計画されたダウンタイムがある場合に、プライマリ LDAP サーバからフェールオーバするセカンダリ LDAP 検索サーバを指定できます。

URL セットの状態は、次のステータスを通じて監視されます。

- INITIAL − LDAP ユーザ認証が設定されていないことを示す。
- RESET − Adaptive Server の管理コマンドで URL が入力されていることを示す。
- READY − URL が接続を受け入れる準備ができていることを示す。
- ACTIVE − URL で LDAP ユーザ認証が成功したことを示す。
- FAILED − LDAP サーバへの接続中に問題が発生したことを示す。
- SUSPENDED − URL がメンテナンス・モードになっており、使用されないことを示す。

次の手順で、フェールオーバと手動によるフェールバックについて説明します。

1　プライマリおよびセカンダリの URL セットが設定されて READY ステータスになります。

2　接続が、プライマリ・サーバ・インフラストラクチャを使用して認証されます。

3　プライマリ・サーバで障害が発生すると、ステータスが FAILED に変わります。

4　セカンダリ・サーバ・インフラストラクチャによる認証が接続で自動的に開始されます。

5　LDAP 管理者によってプライマリ・サーバが修復されて、オンラインに戻ります。Adaptive Server 管理者によりプライマリ LDAP サーバのステータスが READY に変更されます。

6　新しい接続が、プライマリ・サーバを使用して認証されます。

注意　Adaptive Server がセカンダリ LDAP サーバにフェールオーバしたら、データベース管理者は、プライマリ LDAP サーバを手動でアクティブにしてから使用する必要があります。

Adaptive Server で LDAP サーバへの接続時にエラーが発生した場合は、認証が 3 回再試行されます。エラーが続く場合、LDAP サーバのステータスは FAILED になります。Adaptive Server で再試行ループが発生する原因となる LDAP エラーについては、「LDAP ユーザ認証エラーのトラブルシューティング」(132 ページ) を参照してください。

セカンダリ検索 LDAP サーバを設定するには、sp_ldapadmin を使用します。

- セカンダリ DN 検索 URL を設定するには、次のように入力します。

```
sp_ldapadmin set_secondary_dn_lookup_url, <URL>
```

- セカンダリ DN 検索 URL の管理アクセス・アカウントを設定するには、次のように入力します。

  ```
  sp_ldapadmin set_secondary_access_acct, <DN>, <password>
  ```

- 認証のためにプライマリまたはセカンダリ URL の使用をサスペンドするには、次のように入力します。

  ```
  sp_ldapadmin suspend, {primary | secondary}
  ```

- 認証のためにプライマリまたはセカンダリ URL のセットをアクティブ化するには、次のように入力します。

  ```
  sp_ldapadmin activate, {primary | secondary}
  ```

- プライマリおよびセカンダリ LDAP サーバの設定およびステータスの詳細を表示するには、次のように入力します。

  ```
  sp_ldapadmin list
  ```

  sp_ldapadmin list は、list_access_acct および list_urls からの前回の出力を結合します。プライマリ・サーバとセカンダリ・サーバでは次が出力されます。

  - 検索 URL
  - 識別名検索 URL
  - アクセス・アカウント DN
  - アクティブ [true | false]
  - ステータス [ready | active | failed | suspended | reset]

Adaptive Server バージョン 12.5.4 以降には、セカンダリ・サーバをサポートする、次の sp_ldapadmin オプションが用意されています。

- セカンダリ・サーバの DN 検索 URL を表示するには、次のように入力します。

  ```
  sp_ldapadmin list_urls
  ```

- セカンダリ DN 検索の管理アクセス・アカウントを表示するには、次のように入力します。

  ```
  sp_ldapadmin list_access_acct
  ```

- サブコマンドを表示するには、次のように入力します。

  ```
  sp_ldapadmin help
  ```

LDAP サーバのステータスの移行

表 4-7 ～表 4-12 に、sp_ldapadmin の各コマンドを実行したときの LDAP サーバのステータスの移行を示します。

表 4-7 に、sp_ldapadmin set_URL を実行したときのステータスの移行を示します。ここで set_URL は次のコマンドのいずれかを表します。

- set_dn_lookup_url
- set_primary_url
- set_secondary_dn_lookup_url
- set_secondary_url

表 4-7: sp_ldapadmin set_URL 実行時のステータスの移行

初期状態	最終状態
INITIAL	RESET
RESET	RESET
READY	READY
ACTIVE	RESET
FAILED	RESET
SUSPENDED	RESET

表 4-8 に、sp_ldapadmin suspend を実行したときのステータスの移行を示します。

表 4-8: sp_ldapadmin suspend 実行時のステータスの移行

初期状態	最終状態
INITIAL	エラー
RESET	SUSPENDED
READY	SUSPENDED
ACTIVE	SUSPENDED
FAILED	SUSPENDED
SUSPENDED	SUSPENDED

表 4-9 に、sp_ldapadmin activate を実行したときのステータスの移行を示します。

表 4-9: sp_ldapadmin set activate 実行時のステータスの移行

初期状態	最終状態
INITIAL	エラー
RESET	READY
READY	READY
ACTIVE	ACTIVE
FAILED	READY
SUSPENDED	READY

次の表に、Adaptive Server で暗黙に実行される LDAP サーバのステータスの移行を示します。

表 4-10 に、Adaptive Server を再起動するときのステータスの移行を示します。

表 4-10: Adaptive Server 再起動時のステータスの移行

初期状態	最終状態
INITIAL	INITIAL
RESET	RESET
READY	READY
ACTIVE	READY
FAILED	FAILED
SUSPENDED	SUSPENDED

Adaptive Server は、LDAP サーバが READY または ACTIVE ステータスの場合にのみ LDAP ログインを実行します。表 4-11 は、ステータスの移行を示します。

表 4-11: LDAP ログイン成功時のステータスの移行

初期状態	最終状態
READY	ACTIVE
ACTIVE	ACTIVE

表 4-12 に LDAP ログインが失敗した場合のステータスの移行を示します。

表 4-12: LDAP ログイン失敗時のステータスの移行

初期状態	最終状態
READY	FAILED
ACTIVE	FAILED

LDAP ユーザ認証のチューニング

Adaptive Server のオプションの設定とチューニングは、着信接続の負荷および Adaptive Server-LDAP サーバ・インフラストラクチャに基づいて行います。同時着信要求の数に基づいて、以下のオプションを設定します。

- sp_configure を使用して、エンジンあたりのネイティブ・スレッド数を指定する max native threads を設定する。
- sp_ldapadmin を使用して、エンジンあたりの LDAP ユーザ認証ネイティブ・スレッド数を指定する max_ldapua_native_threads を設定する。

ネットワークおよび Adaptive Server/LDAP サーバ・インフラストラクチャの状態に基づいて、(LDAP サーバのバインドおよび検索タイムアウトを指定する) set_timeout オプションを設定します。

set_abandon_ldapua_when_full オプションを設定して、着信接続が max_ldapua_native_threads に達した場合の Adaptive Server の動作を指定します。

パフォーマンスを向上させるように LDAP サーバを設定するには、以下の sp_ldapadmin オプションを使用します。

- set_max_ldapua_desc － LDAPUA 接続要求の同時実行性を管理します。識別名アルゴリズムを使用している場合に、set_max_ldapua_desc の値を大きくすると、Adaptive Server による LDAPUA 接続の処理が高速化します。
- set_num_retries － 試行回数を設定します。この値は、Adaptive Server と LDAP サーバとの間の一時的なエラーの数に基づいて調整します。再試行回数を設定すると、一時的なエラーを取り消すことができます。
- set_log_interval － Adaptive Server のエラー・ログに診断目的で送信されるメッセージの数を制御します。小さい値を指定すると、エラー・ログがメッセージで混雑しますが、特定のエラーを調べるときに効果的です。大きい値を指定すると、エラー・ログに送信されるメッセージの数が少なくなりますが、エラーを調べるための効果は低くなります。set_log_interval は、エラー・ログのサイズに合わせて調整します。

ログイン・マッピングに対する制御の強化

sp_maplogin を使用して、LDAP または PAM で認証されるユーザをローカルの Adaptive Server ログインにマップします。

注意 Kerberos で認証されたユーザをマップするには、sp_maplogin ではなく、sybmapname を使用します。

---

sp_maplogin を使用してログイン・マッピングを作成または変更できるのは、sso_role を持っているユーザだけです。

Adaptive Server では、ログインの認証メカニズム設定とログインを使用するマッピング間の競合が回避されます。潜在的なマッピングの競合は、sp_maplogin ストアド・プロシージャ、alter login コマンド、または create login コマンドによって検出されます。

これらのコントロールでは、以下のマッピングは許可されていません。

- 1 つの Adaptive Server ログイン名から別のログイン名へのマッピング
- ローカルのログインとして既に存在している外部名からのマッピング
- 存在しないログイン名へのマッピング

また、マッピングを使用して認証メカニズムが指定されている場合、メカニズムはターゲットのログインに設定されている認証メカニズムによりチェックされます。

ターゲットのログインの認証メカニズムによって、特定の認証メカニズムを使用するようにログインが制限されている場合は、マッピングで指定されたメカニズムはログインに指定されているメカニズムに一致するか、“ANY” 認証メカニズムと一致する必要があります。

sp_maplogin で、競合が存在することが検出されると、sp_maplogin は失敗し、競合を特定するエラーがレポートされます。

同様に、alter login および create login は、ユーザ・ログインの authenticate with オプションと競合する可能性がある既存のマッピングをチェックします。alter login または create login で競合が検出されると、ログイン・マッピングとの競合を特定するためのエラーがレポートされます。

例

例 1 LDAP ユーザを Adaptive Server の “sa” ログインにマップします。ある企業は、すべてのユーザ・アカウントに対するレポジトリとして LDAP を採用しており、数百台の Adaptive Server を管理できるデータベース管理者 “adminA” および “adminB” を含むすべてのユーザに LDAP 認証を要求するセキュリティ・ポリシーを使用しています。監査は有効になっており、ログイン・イベントは、監査証跡に記録されます。

これらの管理アカウントを “sa” にマップするには、次のように入力します。

```
sp_maplogin LDAP, 'adminA', 'sa'
go
sp_maplogin LDAP, 'adminB', 'sa'
go
```

次のように入力して、LDAP 認証を使用した認証をすべてのユーザに対して要求します。

```
sp_configure 'enable ldap user auth', 2
go
```

“adminA” が Adaptive Server へのログイン中に認証されると、“sa” だけでなく “adminA” に関連付けられた識別名がログイン監査イベントに記録されます。これにより、アクションを実行している各ユーザを監査証跡で識別することができます。

“adminA” および “adminB” のパスワードが LDAP サーバで設定されている場合は、管理対象のすべての Adaptive Server で “sa” パスワードを維持する必要はありません。

この例では、外部の異なる ID やパスワードを認証に使用することもできますが、Adaptive Server 内でこれを行うには、“sa” アカウントに関連付けられた特殊な権限も必要です。

例 2 PAM および LDAP の両方を使用してアプリケーション・ログインにユーザをマップします。ある企業は、PAM および LDAP 認証の両方を採用していますが、それぞれ別の目的で使用しています。会社のセキュリティ・ポリシーでは、LDAP を一般的なユーザ・アカウントの認証メカニズムとして定義し、PAM を中間層アプリケーションなどの特殊なユーザ用として定義しています。中間層アプリケーションは、Adaptive Server への接続プールを設定して、中間層アプリケーションのユーザに代わって要求を処理する場合があります。

LDAP および PAM 両方のユーザ認証のための Adaptive Server の設定は、次のように行います。

```
sp_configure 'enable ldap user auth', 2
go
sp_configure 'enable pam user auth', 2
go
```

Adaptive Server のログイン appX を、中間層アプリケーションに適したパーミッションを使用してローカルに設定します。

```
create login appX with password myPassword
go
alter login appX authenticate with PAM
go
```

単純なパスワードを “appX” にハードコードしていくつかの異なる Adaptive Server でそのパスワードを統一して管理するのではなく、中間層アプリケーションを検証するための追加の情報を使用して中央レポジトリでアプリケーションを認証するカスタムの PAM モジュールを開発します。

クライアント・アプリケーションのログイン “appY” には、LDAP ID とパスワードによるユーザの LDAP 認証が必要です。すべての LDAP 認証ユーザをログイン “appY” にマップするには、sp_maplogin を使用します。

```
create login appY with password myPassword
go
sp_maplogin LDAP, NULL, 'appY'
go
```

“appY” のユーザは会社の ID とパスワードを使用して認証されてから、ローカルの Adaptive Server のログイン “appY” にマップされ、データベース・アクションを実行します。LDAP ユーザの ID を使用して認証が行われると、監査証跡に記録され、アプリケーションのログイン “appY” に適したパーミッションで実行されます。

外部認証のログイン・マッピング

外部認証メカニズムを設定したときに、内部 Adaptive Server ログインに対する外部ユーザのマッピングが 1 つだけあり、マッピングが正常に認証された場合、Adaptive Server は外部ユーザのパスワードと一致するように内部ログインのパスワードを更新します。次に例を示します。

1　ユーザには、Adaptive Server のログイン名 user_ase (パスワードは user_password) と、LDAP ログイン名 user_ldap (パスワードは user_ldappasswd) があります。

   Adaptive Server では、user_ldap と user_ase が一対一でマッピングされています。

2 user_ldap が user_ldappassword を使用して Adaptive Server にログインすると、Adaptive Server は user_ase のパスワードを user_ldappassword に更新します。

Adaptive Server のログイン名を LDAP のパスワードにマッピングする利点は、LDAP サーバがクラッシュした場合に、ユーザが最も最近使用した LDAP パスワードでログインできることです。つまり、ユーザには Adaptive Server 認証についてユーザ名と LDAP パスワードの 1 対 1 のマッピングがあり、ログイン認証時にパスワードを使用するとそのパスワードがローカルで更新されるため、ユーザが Adaptive Server に対して継続的に認証されているように見えます。

ただし、複数のユーザがローカル・ユーザにマッピングされていると、パスワードはローカルに更新されません。LDAP サーバがクラッシュした場合、Adaptive Server では 1 人の Adaptive Server ユーザにマッピングされている複数の外部ユーザの認証は実行されません。

LDAP ユーザ認証エラーのトラブルシューティング

Adaptive Server では、LDAP サーバと通信中に次のような一時的なエラーが発生する場合があります。通常、接続を再試行するとこれらのエラーは解決します。再接続した後も同様のエラーが解決しない場合は、Adaptive Server によって LDAP サーバに FAILED のステータスが設定されます。

- LDAP_BUSY − サーバがビジー。
- LDAP_CONNECT_ERROR − 接続中のエラー。
- LDAP_LOCAL_ERROR − クライアント側のエラー。
- LDAP_NO_MEMORY − クライアント側にメモリを割り付けることができない。
- LDAP_OPERATIONS_ERROR − サーバ側のエラー。
- LDAP_OTHER − 不明なエラー・コード。
- LDAP_ADMINLIMIT_EXCEEDED − 検索が制限を超えている。
- LDAP_UNAVAILABLE − サーバが要求を処理できない。
- LDAP_UNWILLING_TO_PERFORM − サーバが要求を処理しない。
- LDAP_LOOP_DETECT − 参照中にループが検出された。
- LDAP_SERVER_DOWN − サーバに到達できない (接続が失敗した)。
- LDAP_TIMEOUT − ユーザ指定の時間内にオペレーションが完了しないために LDAP API が失敗した。

一時的なエラーや多数の同時ログイン要求によって、エラー・ログで大量のエラー・メッセージが繰り返される場合があります。ログを読みやすくするために、次のエラー・メッセージ・ログ・アルゴリズムが使用されます。

1　初めてログに記録されるメッセージは、そのまま記録されます。

2　メッセージが最後に記録されてから 3 分を超えた場合は、次のようになります。

- エラー・メッセージが記録される。
- メッセージが最後に出力されてからメッセージが繰り返された回数が記録される。
- メッセージが出力されてから経過した時間が分単位で記録される。

次の原因で発生した認証エラーは、LDAP エラーとは見なされず、認証要求を再試行する条件にはなりません。

- 不正なパスワードまたは無効な識別名によるバインド・エラー。
- 0 の結果セットを返すか、属性値を返さない、バインドが成功した後の検索。

URL 解析中に検出される構文エラーは LDAP URL の設定時にキャッチされるため、上記のいずれのカテゴリにも該当しません。

LDAP サーバの設定

LDAP (Lightweight Directory Access Protocol) のユーザ認証では、SSL/TLS (Secure Sockets Layer/Transport Layer Security) プロトコルがサポートされており、Adaptive Server と LDAP サーバ間でのデータ転送の安全性を確保できます。

❖ LDAP サーバへの接続の設定

1　信頼されたルート証明書がすべて同じファイルに保存されていることを確認します。

信頼されたサーバを定義すると、Adaptive Server によってセキュア接続が次のように設定されます。ここで、*servername* は現在の Adaptive Server の名前です。

- *$SYBASE_CERTDIR* を定義した場合は、Adaptive Server によって *$SYBASE_CERTDIR/servername.txt* (UNIX の場合) または *%SYBASE_CERTDIR%\servername.txt* (Windows の場合) から証明書がロードされます。
- *$SYBASE_CERTDIR* を定義しなかった場合は、Adaptive Server によって *$SYBASE/$SYBASE_ASE/certificates/servername.txt* (UNIX の場合) または *%SYBASE%\%SYBASE_ASE%\certificates\servername.txt* (Windows の場合) から証明書がロードされます。

2　Adaptive Server を再起動することによって、信頼されたルート証明書ファイルを変更します。

3 sp_ldapadmin を使用し、*ldap://* 形式の URL ではなく *ldaps://* 形式の URL を指定して、LDAP サーバのセキュア・ポートへのセキュア接続を確立します。

4 次のいずれかの構文を使用して、プレイン・テキストでの TCP 接続を介して TLS セッションを確立します。

```
sp_ldapadmin 'starttls_on_primary', {true | false}
```

または

```
sp_ldapadmin 'starttls_on_secondary', {true | false}
```

注意 LDAP サーバ接続には connect timeout オプションがありません。LDAP サーバが応答を停止した場合は、すべてのログイン接続も応答を停止します。

LDAPS ユーザ認証の強化

以前のバージョンの Adaptive Server では、CA (認証局) によって信頼されたルート・ファイルに変更を加えた場合に、Adaptive Server を再起動して変更を有効にする必要があります。Adaptive Server バージョン 15.0.3 以降では、信頼されたルート・ファイルへの変更がサポートされているため、サーバを再起動する必要がありません。新しく追加されたサブコマンド reinit_descriptors は、LDAP サーバ記述子のバインドを解除して、ユーザ認証サブシステムを再初期化します。このオプションの構文については、『リファレンス・マニュアル：プロシージャ』を参照してください。

- このコマンドを実行するには、システム・セキュリティ担当者のパーミッションが必要です。
- システム・セキュリティ担当者のパーミッションを持つユーザが、このコマンドを実行しないで、信頼されたルート・ファイルを変更した場合、ハウスキーピング・ユーティリティのチョア・タスクでは、ユーザ認証サブシステムを 60 分ごとに再初期化するように設計された、新しいチョアが使用されます。

自動的な LDAP ユーザ認証とフェールバック

Adaptive Server 15.0.3 では、セカンダリ LDAP サーバがサポートされています。以前のバージョンでは、障害の発生したプライマリ LDAP サーバをオンライン状態にしたら、新しい LDAP ログインを認証して、プライマリ LDAP サーバに移動するために、LDAP サーバを手動でアクティブにする必要がありました。

バージョン 15.0.3 以降では、LDAP サーバを自動的にアクティブにするための、新しいチョア 'set_failback_interval' が Adaptive Server のハウスキーピング・ユーティリティに追加されています。構文については、「LDAP フェールバック時間間隔の設定」(135 ページ) を参照してください。

sp_ldapadmin set_failback_interval の set_failback_interval オプションは、障害の発生した LDAP サーバをアクティブにするための試行間隔を設定します。このパラメータを設定しない場合は、デフォルト値である 15 分が使用されます。『リファレンス・マニュアル：プロシージャ』の「sp_ldapadmin」を参照してください。

プライマリ URL のステータスが FAILED の場合、ハウスキーピング・タスクは、プライマリ・アクセス・アカウントの DN (識別名) とパスワードを使用して、プライマリ URL をアクティブにしようとします。プライマリ・アクセス・アカウントを設定していない場合、ハウスキーピング・タスクは匿名バインドの使用を試みます。初回の試行時にバインド操作が失敗した場合、ハウスキーピング・タスクは、設定された再試行回数だけバインド操作を再試行します。バインド操作が成功すると、プライマリ URL のステータスが READY になります。

セカンダリ URL のステータスが FAILED の場合、ハウスキーピング・タスクは、同様の方法でセカンダリ URL をアクティブにしようとします。

sp_ldapadmin の reinit_descriptors オプションは、証明書ファイルが変更されたときに実行されます。この場合、LDAP ユーザ認証サブシステムは 60 分ごとに再初期化されます。

フェールバック間隔がユーザによって設定されると、ハウスキーピング・タスクは、チョアを一掃するたびに、障害の発生した LDAP サーバの有無を調べます。障害の発生した LDAP サーバが見つかった場合は、フェールバック時間間隔で指定した時間が経過すると、LDAP サーバのアクティブ化が試みられます。

LDAP フェールバック時間間隔の設定

sp_ldapadmin set_failback_interval の構文は次のとおりです。ここで、*time_in_minutes* は -1 ～ 1440 分 (24 時間) の値です。

```
sp_ldapadmin 'set_failback_interval', time_in_minutes
```

- 値 0 は、フェールバックが手動であることを示します。つまり、ハウスキーピング・タスクによる LDAP サーバの自動フェールバックは試みられません。ユーザはこのタスクを手動で実行する必要があります。
- この値を -1 にすると、フェールオーバ時間間隔が、デフォルト値である 15 分に設定されます。
- パラメータを使用しないで sp_ldapadmin 'set_failback_interval' を発行した場合、sp_ldapadmin はフェールバック間隔の設定値を表示します。

- パラメータを使用しないで sp_ldapadmin を発行した場合、sp_ldapadmin の出力には、フェールバック時間間隔が次のように示されます。

```
 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)
```

例

この例では、LDAP フェールバック時間間隔が 60 分に設定されます。

```
sp_ldapadmin 'set_failback_interval' 60
```

この例では、LDAP フェールバック

時間間隔がデフォルト値である 15 分に設定されます。

```
sp_ldapadmin 'set_failback_interval' -1
```

この例では、フェールバック間隔の設定値が表示されます。

```
sp_ldapadmin 'set_failback_interval'
```

```
The LDAP property 'set_failback_interval' is set to '15
    minutes'.
```

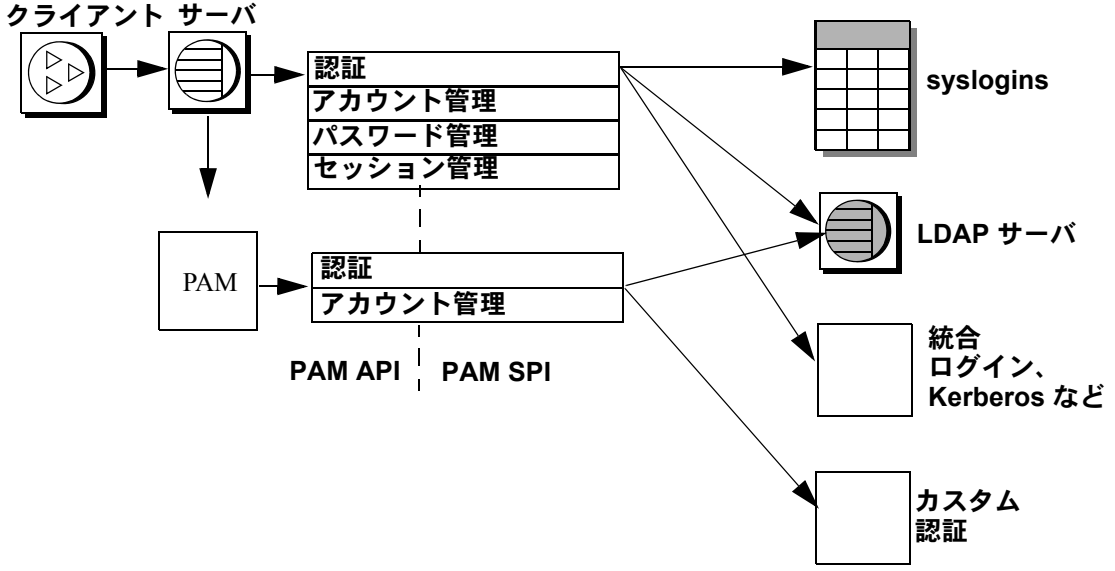
PAM を使用する認証のための Adaptive Server の設定

PAM (Pluggable Authentication Module) のサポートにより、認証を必要とするアプリケーションを変更せずに、複数の認証サービス・モジュールをまとめて使用できます。

PAM により Adaptive Server が Solaris や Linux のオペレーティング・システムに統合され、ユーザ・アカウントや認証メカニズムの管理が単純化され、総保有コスト (TCO) が削減されます。ユーザは、独自の認証モジュールや許可モジュールをカスタマイズしたり、作成したりできます。

注意 現在 PAM がサポートされているプラットフォームは Linux と Solaris です。PAM ユーザ認証の詳細については、各オペレーティング・システムのマニュアルを参照してください。

図 4-2: PAM アーキテクチャ



Adaptive Server は、ログイン・パケットから取得したログイン名とクレデンシャルを PAM API に渡します。PAM は、オペレーティング・システムの設定ファイルの指定に従ってサービス・プロバイダ・モジュールをロードし、認証プロセスを完了するための関数を呼び出します。

Adaptive Server での PAM の有効化

Linux と Solaris には定義済みの PAM モジュールがあります。これらのモジュールのいずれか一方を使用することも、独自のモジュールを作成することもできます。独自のモジュールを作成する場合は、オペレーティング・システムのマニュアルに記載されている PAM モジュールの作成に関する指示に従ってください。

注意 PAM モジュールを作成する場合は、RFC 86.0 “Unified Login With Pluggable Authentication Modules (PAM)” に準拠する必要があります。Adaptive Server では、RFC の認証管理モジュールがサポートされています。アカウント管理、セッション管理、またはパスワード管理のモジュールはサポートされていません。

オペレーティング・システムの設定

PAM サポートを有効にするには、各オペレーティング・システムを次のように設定します。

- Solaris では、*/etc/pam.conf* に次の行を追加します。

```
ase auth required /user/lib/security/$ISA/pam_unix.so.1
```

- Linux では、*/etc/pam.d/ase* という新しいファイルを作成して次の行を入力します。

```
auth requried /lib/security/pam_unix.so
```

これらのエントリの作成方法の詳細については、オペレーティング・システムのマニュアルを参照してください。

同一マシンでの 32 ビット・サーバと 64 ビット・サーバの実行

$ISA は、32 ビット・ライブラリと 64 ビット・ライブラリを同時に実行するために使用される環境変数です。

Solaris の 32 ビット・マシンでは $ISA は空文字列に置き換えられ、64 ビット・マシンでは文字列 “sparcv9” に置き換えられます。

32 ビット・サーバと 64 ビット・サーバの両方を使用する場合は、32 ビット版 PAM モジュールを任意のディレクトリに格納し、64 ビット版 PAM モジュールをそのディレクトリのサブディレクトリに格納します。

pam.conf のエントリは次のようになります。

```
$ 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
```

注意　*pam.conf* に指定できる変数は $ISA のみです。

PAM ユーザ認証のための Adaptive Server の設定

enable pam user auth は、PAM ユーザ認証サポートを有効にします。

```
sp_configure "enable pam user auth", 0 | 1 | 2
```

各パラメータの意味は、次のとおりです。

- 0 － PAM 認証を無効にします。これがデフォルト値です。
- 1 － Adaptive Server は最初に PAM 認証を試行し、失敗した場合は syslogins 認証を使用します。
- 2 － PAM 認証のみを使用できるように指定します。

注意　PAM が有効な場合、パスワード管理は PAM サービス・プロバイダに委任されます。

Adaptive Server ログインと PAM ユーザ・アカウント

enable PAM user authentication を設定し、Adaptive Server とオペレーティング・システムの両方で PAM を設定したら、ユーザ・アカウントを設定します。オペレーティング・システム管理者またはネットワーク・セキュリティ管理者が PAM サービス・プロバイダのユーザ・アカウントの作成と管理を行い、データベース管理者が Adaptive Server のアカウントの作成と管理を行います。また、データベース管理者は、管理オプションを使用して、Adaptive Server と PAM サーバなどの外部認証メカニズムを統合するときのログイン・アカウントを柔軟に設定できます。データベース管理者は、従来のコマンドとプロシージャを使用して、Adaptive Server アカウントの役割、デフォルト・データベース、デフォルト言語、およびその他のログイン固有の属性の管理を続行できます。

表 4-13 は、ログイン時の syslogins テーブルの変更を示します。ここに示す変更は、PAM ユーザ認証が設定済みで、ログインが PAM の使用を制限されておらず、create login マッピングを設定していないことを前提としています。

表 4-13: PAM による syslogins の変更

syslogins にそのユーザのローが既に存在する	PAM 認証に成功	syslogins の変更
いいえ	はい	変更なし、ログインは失敗
いいえ	いいえ	変更なし、ログインは失敗
はい	はい	パスワードが変更された場合は、ローが更新される
はい	いいえ	変更なし

機能拡張されたログイン制御

前述の LDAP と PAM の項目で説明した方法に従って、サーバ全体の認証メカニズムを使用するように Adaptive Server を設定します。また、以下に説明する、機能拡張された Adaptive Server ログイン制御を使用して、サーバ上のログインごとに特定の認証メカニズムを使用するように指定することもできます。

ログインごとの制御は、サーバの認証メカニズムを移行中である場合や、ローカルなサーバ管理が必要で、集中管理されたユーザ・ログインに関連付けられていないサーバ固有のログインを制御する場合に便利です。

認証の強制

alter login と create login に対して以下のパラメータを使用して、ログインで特定の認証プロセスを使用するように強制できます。

- ASE − syslogins テーブルに格納されているパスワードを使用する Adaptive Server 内部認証を使用する。
- LDAP − LDAP サーバによる外部認証を使用する。
- PAM − PAM による外部認証を使用する。
- ANY − デフォルトのユーザ認証メソッド。ユーザに対して ANY 認証を指定すると、Adaptive Server は外部認証メカニズムが定義されているかどうかを調べます。定義されている場合は、そのメカニズムが使用されます。定義されていない場合は、Adaptive Server の認証が使用されます。

Adaptive Server は次の順序で外部認証メカニズムを調べます。

1 LDAP

2 PAM (Pluggable Authentication Module)。LDAP と PAM の両方が有効な場合、ユーザに対して PAM 認証は試行されない。

3 PAM と LDAP がどちらも有効になっていない場合は、syslogins によってログインが認証される。

“sa” などのログイン・アカウントは、引き続き syslogins カタログを使用して検証されます。ログインの認証を設定できるのは、SSO の役割を付与されているユーザだけです。

alter login を使用してログインを認証する例を次に示します。

```
alter login nightlyjob modify authenticate with ASE
sp_displaylogin "nightlyjob"
```

これによって次のような出力が表示されます。

```
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
```

sp_maplogin を使用したログインのマッピング

次の構文で sp_maplogin を使用してログインをマップできます。

sp_maplogin (authentication_mech | null),
(*client_username* | null), (*action* | *login_name* | null)

各要素の意味は次のとおりです。

- authentication_mech － sp_maplogin の authenticate with オプションに指定できる有効な値の 1 つ。
- *client_username* － 外部ユーザ名。オペレーティング・システム名、LDAP サーバのユーザ名、または PAM ライブラリが認識できる任意の名前を指定できます。null 値を指定すると、すべてのログイン名が有効になります。
- *action* － create login または drop を指定します。create login を使用すると、認証時にログインが作成されます。drop はログインを削除するときに使用します。
- *login_name* は、syslogins に既に存在する Adaptive Server ログインです。

次の例では、外部ユーザ “jsmith” を Adaptive Server ユーザ “guest” にマップします。認証されると、“jsmith” には “guest” の特権が付与されます。監査ログイン・レコードには、*client_username* と Adaptive Server ユーザ名の両方が示されます。

```
sp_maplogin NULL, "jsmith", "guest"
```

次の例は、LDAP で認証されたすべての外部ユーザについて、ログインが存在しない場合は新規ログインを作成するように Adaptive Server に指示します。

```
sp_maplogin LDAP, NULL, "create login"
```

マッピング情報の表示

sp_helpmaplogin はマッピング情報を表示します。

```
sp_helpmaplogin [ (authentication_mech | null), (client_username | null) ]
```

各パラメータの意味は、次のとおりです。

- *client_username* − 外部ユーザ名です。

パラメータを指定せずに sp_helpmaplogin を使用した場合、Adaptive Server に現在ログインしているすべてのユーザに関するログイン情報が表示されます。上記のパラメータを使用すると、出力をクライアント・ユーザ名または認証メカニズムの特定のセットに限定できます。

次に、すべてのログインに関する情報を表示する例を示します。

```
sp_helpmaplogin

authentication   client name   login name
--------------   -----------   ------------------
NULL             jsmith        guest
LDAP             NULL          create login
```

認証メカニズムの設定

Adaptive Server で使用する認証メカニズムを設定するには、*@@authmech* グローバル変数を使用します。

たとえば、Adaptive Server でフェールオーバ対応の LDAP ユーザ認証が有効になっており (enable ldap user auth = 2) 、ユーザ “Joe” が ANY 認証の外部ユーザである場合、Joe がログインすると、Adaptive Server は LDAP ユーザ認証で Joe を認証しようとします。Joe の LDAP でのユーザ認証が失敗すると、Adaptive Server は Adaptive Server 認証を使用して Joe を認証します。これが成功するとログインできます。

この場合の *@@authmech* グローバル変数の値は次のとおりです。

```
select @@authmech
---------------------------------
ase
```

Adaptive Server が厳密な LDAP ユーザ認証を使用するように設定されており (enable ldap user auth = 2)、Joe が有効なユーザとして LDAP に追加された場合、Joe がログインするときの @@authmech の値は次のようになります。

```
select @@authmech
----------------------------------
ldap
```

第 5 章

役割の管理

この章では、Adaptive Server での役割の使用方法について説明します。

ユーザに対する役割の作成と割り当て

役割とは、役割の割り付け対象者が自身のジョブを実行できるようにするための特権の集まりです。Adaptive Server でサポートされる役割を使用すると、各ユーザの責任範囲を指定することができます。Adaptive Server には、システム管理者やシステム・セキュリティ担当者などのシステム標準の役割と、システム・セキュリティ担当者が作成し、ユーザ、ログイン・プロファイル、またはその他の役割に付与された役割であるユーザ定義の役割があります。オブジェクト所有者は、必要に応じて、特定の役割にデータベース・アクセス権を付与できます。

データベース・ユーザ追加の手順の最後に、必要に応じてユーザに特別な役割を割り当て、パーミッションを付与します。パーミッションの詳細については、「第 6 章 ユーザ・パーミッションの管理」を参照してください。

システム標準の役割

表 5-1 は、システム標準の役割、grant role コマンドまたは revoke role コマンドの *role_granted* オプションに使用する値、その役割を持つユーザによって一般に実行されるタスクを示します。

注意 それぞれの役割についての詳細は、以降の項を参照してください。

表 5-1: システム標準の役割と関連するタスク

役割	*role_granted* の値	説明
システム管理者	sa_role	Adaptive Server のデータベースとディスク記憶領域の管理と維持
システム・セキュリティ担当者	sso_role	セキュリティ関連タスクの実行
オペレータ	oper_role	サーバワイドのデータベースのバックアップとロード
Sybase サポート・センタ	sybase_ts_role	データベース構造の分析と修復
複写	replication_role	ユーザ・データの複写
分散トランザクション管理	dtm_tm_role	サーバ間のトランザクションのコーディネート
高可用性	ha_role	フェールオーバの管理と実行
モニタリングと診断	mon_role	パフォーマンスと診断のモニタリングの管理と実行
Job Scheduler 管理	js_admin_role	Job Scheduler の管理
Job Scheduler ユーザ	js_user_role、 js_client_role	Job Scheduler によるジョブの作成と実行
リアルタイム・メッセージング	messaging_role	リアルタイム・メッセージングの管理と実行
Web サービス	webservices_role	Web Services の管理
キー管理者	keycustodian_role	暗号化キーの作成および管理

注意 sa_role は sa_role を持つユーザによって付与されます。その他すべてのシステム標準の役割は、sso_role を持つユーザが生成できます。ユーザ定義の役割が sa_role とその他のシステム標準の役割の両方に付与されている場合、この役割は、sa_role と sso_role の両方を持つユーザによってのみ付与されます。

システム管理者の権限

システム管理者は以下のことを行います。

- アプリケーションに固有ではないタスクの処理
- Adaptive Server の任意アクセス制御システムの外部での作業

システム管理者の役割は、通常は特定の Adaptive Server ログインに付与されます。そのユーザが実行する処理はすべてそのユーザのサーバ・ユーザ ID で追跡できます。サーバ管理の作業量が 1 人で実行できる程度であれば、個人のログインではなく、Adaptive Server のインストール時に作成される “sa” アカウントを使用することもできます。インストール時に、“sa” アカウントのユーザは、システム管理者の役割、システム・セキュリティ担当者の役割、オペレータの役割を使用できることになります。“sa” アカウントのパスワードを知っていれば誰でも、そのアカウントにログインしてこれらの役割を持つことができます。

システム管理者が保護システムの外部で作業することは、安全対策の 1 つとなります。たとえば、データベース所有者が sysusers テーブル内のすべてのエントリを誤って削除してしまった場合でも、バックアップがあれば、システム管理者がそのテーブルをリストアできます。コマンドの中には、システム管理者しか発行できないものもあります。システム管理者しか発行できないコマンドは、disk init、disk refit、disk reinit、shutdown、kill、disk mirror、mount、unmount、および複数のモニタリングを行うコマンドです。

パーミッションを付与するとき、システム管理者はオブジェクト所有者として扱われます。システム管理者が、別のユーザのオブジェクトに対するパーミッションを付与すると、sysprotects と sp_helprotect の出力では、オブジェクト所有者の名前が付与者として表示されます。

システム管理者は、データベースにログインするときに、データベース所有者の ID を自動的に想定し、すべてのデータベース所有者の権限を使用します。この自動マッピングは、ユーザに割り当てられたエイリアスに関係なく実行されます。システム管理者は、dbcc コマンド、診断機能、データ・ページの読み取り、データやインデックスのリカバリなど、通常、データベース所有者用に予約されているタスクを実行できます。

システム・セキュリティ担当者の権限

システム・セキュリティ担当者は、Adaptive Server のセキュリティに関係する作業を実行します。これらの作業には、次のものがあります。

- システム・セキュリティ担当者、オペレータ、およびキー管理者の役割の付与
- 監査システムの管理
- パスワードの変更
- 新しいログインの追加
- ログインの削除
- ログイン・アカウントのロックとロック解除
- ユーザ定義の役割の作成と付与
- ネットワークベース・セキュリティの管理
- set proxy コマンドまたは set session authorization コマンドを使用するためのパーミッション付与
- ログイン・プロファイルの作成
- 暗号化の管理

システム・セキュリティ担当者は、監査を有効にする必要があるため、すべてのデータベースにアクセスできますが、通常はデータベース・オブジェクトに対する特別なパーミッション (暗号化キーと暗号化カラムの decrypt パーミッションを除きます。『暗号化カラム・ユーザーズ・ガイド』を参照してください) は持ちません。sybsecurity データベースは例外で、このデータベースの sysaudits テーブルにはシステム・セキュリティ担当者以外はアクセスできません。システム・セキュリティ担当者しか実行できないシステム・プロシージャもあります。

システム・セキュリティ担当者は、ユーザの不注意による保護システムの変更を修復できます。たとえば、データベース所有者がパスワードを忘れた場合、システム・セキュリティ担当者はパスワードを変更してデータベース所有者がログインできるようにします。

システム・セキュリティ担当者は、システム管理者とログインの管理責任を共有します。システム・セキュリティ担当者は、ログインとログイン・プロファイルの管理を担当します。

システム・セキュリティ担当者は、sa_role を除く、すべてのシステム標準の役割を付与できます。ユーザ定義の役割を作成して、その役割をユーザ、他の役割、ログイン・プロファイル、またはグループに付与することもできます。「ユーザに対する役割の作成と割り当て」(145 ページ) を参照してください。

オペレータの権限

オペレータの役割を付与されたユーザは、個々のデータベースの所有者にならなくても、サーバワイドでデータベースのバックアップとリストアを実行できます。オペレータの役割を付与されているユーザは、すべてのデータベースに対して次のコマンドを使用できます。

- dump database
- dump transaction
- load database
- load transaction
- checkpoint
- online database

Sybase サポート・センタ

Sybase 製品の保守契約を結んでいるサポート・センタの技術者は、サポート・センタの役割を使用して、トレース出力、一貫性チェック、データ構造へのパッチを通じて内部メモリ・データ構造とディスク上のデータ構造を表示できます。この役割は、問題の分析とデータのリカバリを手動で行うために使用されます。解決する問題によっては、データにアクセスするためにシステム標準の役割を追加する必要のある操作もあります。このような分析または修復を実行する場合、システム・セキュリティ担当者はこの役割を経験豊富な Sybase 技術者に対してのみ付与することをおすすめします。

複写の役割

Replication Server と ASE Replicator を管理するユーザには、複写の役割が必要です。この役割の詳細については、『Replication Server 管理ガイド』と『ASE Replicator ユーザーズ・ガイド』を参照してください。

分散トランザクション管理の役割

この役割は、分散トランザクション管理 (DTM) トランザクション・コーディネータが、システム・ストアド・プロシージャによるサーバ間のトランザクションの管理を可能にするために使用します。DTM XA インタフェースを使用するクライアントには、この役割が必要です。『Adaptive Server 分散トランザクション管理機能の使用』を参照してください。

高可用性の役割

高可用性の役割は、コマンドとストアド・プロシージャを通じてプライマリ・サーバとコンパニオン・サーバを管理する、高可用性サブシステムを設定するために必要です。『高可用性システムにおける Sybase フェールオーバの使用』を参照してください。

モニタリングと診断

この役割は、Adaptive Server のモニタリング・テーブルを管理するために必要です。モニタリング・テーブルのリモート・プロシージャ・コールの実行やモニタリングされたデータの収集の管理には、この役割が必要です。『パフォーマンス＆チューニング・シリーズ：モニタリング・テーブル』を参照してください。

Job Scheduler の役割

Job Scheduler のオペレーションに対するパーミッションを管理するためのシステム標準の役割には、次の 3 つがあります。

- js_admin_role － Job Scheduler の管理に必要な役割であり、ストアド・プロシージャにアクセスして Job Scheduler の管理操作を修正、削除、実行できます。
- js_user_role － Job Scheduler のストアド・プロシージャを使用してスケジュール・ジョブを作成、修正、削除、実行するために必要な役割です。
- js_client_role － 定義済みジョブを使用できますが、ジョブを作成または変更することはできません。

詳細については、『Job Scheduler ユーザーズ・ガイド』を参照してください。

リアルタイム・メッセージングの役割

msgsend、msgrecv、および一部の sp_msgadmin コマンドを実行するために、リアルタイム・メッセージング・サブシステム (RTMS) で使用されます。詳細については、『Messaging Services ユーザーズ・ガイド』を参照してください。

Web Services の役割

この役割は、Web Services サブシステムで、create service、create existing service、drop service、および alter service コマンドを実行するために使用されます。『Web Services ユーザーズ・ガイド』を参照してください。

キー管理者の役割

キー管理者の役割は、暗号化キーの作成と変更、システム暗号化パスワードの設定、ユーザのキー・コピーの設定などのキー管理の責任があります。『暗号化カラム・ユーザーズ・ガイド』を参照してください。

ユーザ定義の役割の計画

ユーザ定義の役割を実際に使用する前に、次のことを決定します。

- 作成する役割
- 各役割の責任
- 役割の階層における各役割の位置
- 階層内で相互排他的な関係にある役割と、その排他性をメンバシップ・レベルとアクティブ化レベルのどちらで設定するか

名前の重複を避けるには、命名規則に従ってユーザ定義の役割を作成するようにします。たとえば、役割名の末尾には “_role” を付けます。Adaptive Server は、そのような制限についてはチェックしません。

ユーザまたはログイン・プロファイルへ直接付与されるユーザ定義の役割の名前は、いかなるログイン名またはログイン・プロファイル名と重複しないようにします。ある役割をユーザと同じ名前にする必要がある場合には、新しい役割を作成してそれに元の役割を組み込んでから、その新しい役割をユーザに付与することによって、矛盾を避けることができます。

作成する役割とその関係の計画が完了したら、ビジネス要件とユーザの責任に従って役割を割り付ける方法を決定してください。

ユーザがユーザ・セッションごとにアクティブ化できる役割の最大数は 127 です。

サーバワイドで作成できるユーザ定義の役割の最大数は、992 です。

ユーザ定義の役割の作成

sso_role があるユーザは、create role コマンドを使用して役割を作成します。『リファレンス・マニュアル：コマンド』の「create role」を参照してください。

create role コマンドは、 master データベースでのみ使用できます。

パスワードを使用する場合は、その役割をアクティブ化するユーザがパスワードを指定する必要があります。パスワードが指定されている役割が、デフォルトの役割またはログイン・プロファイルに付与される自動アクティブ化役割としてログイン時にアクティブ化される場合、その役割を使用することはできません。

たとえば、パスワードなしで intern_role を作成するには、次のように入力します。

```
create role intern_role
```

doctor_role を作成して、パスワード “physician” を割り当てるには、次のように入力します。

```
create role doctor_role with passwd "physician"
```

ユーザ定義の役割を作成できるのは、システム・セキュリティ担当者だけです。

役割のパスワードの追加と削除

役割のパスワードを追加したり削除したりできるのは、システム・セキュリティ担当者だけです。

システム標準の役割またはユーザ定義の役割のパスワードを追加または削除するには、alter role コマンドを使用します。

```
alter role role_name
[add passwd password | drop passwd]
```

たとえば、oper_role にパスワード “oper8x” が必要となるようにするには、次のように入力します。

```
alter role oper_role add passwd oper8x
```

役割からパスワードを削除するには、次のように入力します。

```
alter role oper_role drop passwd
```

注意 役割にパスワードを割り当てる場合は、その役割をアクティブ化するときに、役割に付与されたユーザが Adaptive Server にパスワードを指定する必要があります。

役割の階層と相互排他性

システム・セキュリティ担当者は、役割の階層を定義できます。これは、ユーザに 1 つの役割が与えられると、階層内のそれより下位の役割もそのユーザに与えられるというものです。役割 role1 を別の役割である role2 に付与する場合は、role2 が role1 を含んでいる階層を設定します。たとえば、役割 “chief_financial_officer” に、“financial_analyst” と “salary_administrator” の両方の役割が含まれるようにします。

chief financial officer は、すべてのタスクを実行でき、salary administrator と financial analyst が参照可能なデータはすべて参照できます。

さらに、役割の相互排他性を定義すると、作業方式の静的または動的な分割を実行できます。次のものについて、役割が相互排他になるように定義できます。

- メンバシップ － 1 人のユーザに 2 つの異なる役割を付与することはできません。たとえば、“payment_requestor” と “payment_approver” の両方の役割が同一ユーザに付与されないようにする場合です。
- アクティブ化 － 1 人のユーザが 2 つの異なる役割をアクティブ化、つまり有効にすることはできません。たとえば、1 人のユーザに “senior_auditor” と “equipment_buyer” の両方の役割が付与されていても、両方の役割を同時には有効にできないようにする場合です。

システム標準の役割は、ユーザ定義の役割と同じく、役割階層内に定義することや、相互排他となるように定義することができます。たとえば、“super_user” という役割に、システム管理者、オペレータ、テクニカル・サポートの各役割が含まれるようにします。役割の分割を実行するには、システム管理者とシステム・セキュリティ担当者の役割が、メンバシップに関して相互排他になるように、つまり、1 人のユーザに両方の役割を付与できないように定義できます。

役割の相互排他性の定義と変更

2 つの役割間の相互排他性を定義するには、次の構文を使用します。

alter role *role1* { add | drop } exclusive { membership | activation } *role2*

たとえば、メンバシップ・レベルで、intern_role と specialist_role が相互排他となるように定義するには、次のように入力します。

```
alter role intern_role add exclusive membership
specialist_role
```

上記の例では、intern_role のメンバシップを持っているユーザが specialist_role のメンバにもならないように制限します。

sso_role と sa_role が、アクティブ化レベルで相互排他となるように定義するには、次のコマンドを入力します。このコマンドは、sso_role と sa_role のメンバであるユーザが、両方の役割を同時に持つことを禁止します。

```
alter role sso_role add exclusive activation sa_role
```

役割の階層の定義と変更

役割の階層を定義するには、初めに階層のタイプと役割を選択し、次に役割を別の役割に付与することによって階層を実装します。

次に例を示します。

```
grant role intern_role to specialist_role
grant role doctor_role to specialist_role
```

“specialist” に “doctor” と “intern” の両方が持つすべての権限を付与します。

役割 “super_user” に、システム標準の役割である sa_role と oper_role が含まれるような階層を作成するには、次のように指定します。

```
grant role sa_role to super_user
grant role oper_role to super_user
```

注意　パスワードの必要な役割が別の役割に含まれている場合、上位の役割が付与されているユーザは、下位の役割を使用するときもパスワードは必要ありません。上記の例では、役割 “doctor” に、通常はパスワードが必要であるとします。役割 “specialist” が付与されているユーザは “doctor” のパスワードを入力する必要はありません。“doctor” は “specialist” に含まれており、役割のパスワードは最高レベルの役割についてのみ要求されるためです。

役割の階層を作成するときは、次の規則に従います。

- ある役割を、それを直接含む別の役割に付与することはできません。これによって、重複が防止されます。

  上記の例では、役割 “doctor” を役割 “specialist” に付与することはできません。“specialist” には “doctor” が既に含まれているためです。

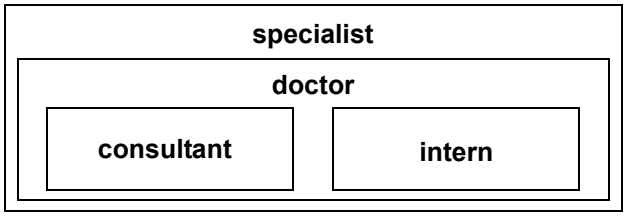
- ある役割を、それを直接含まない別の役割に付与することはできます。

  たとえば、図 5-1 では、役割 “specialist” に役割 “doctor” が既に含まれており、“doctor” に役割 “intern” が含まれていますが、“intern” を “specialist” に付与できます。その後で、“doctor” を “specialist” から削除しても、“specialist” に “intern” が含まれる状態は変わりません。

  図 5-1 では、“doctor” は役割 “consultant” のパーミッションを持っています。これは、“consultant” が “doctor” に付与されているためです。役割 “specialist” にも役割 “consultant” のパーミッションがあります。これは、“specialist” には役割 “doctor” が含まれ、役割 “doctor” には “consultant” が含まれるためです。

  ただし、“intern” には、役割 “consultant” の権限はありません。これは、“intern” には役割 “consultant” が直接的にも間接的にも含まれないためです。

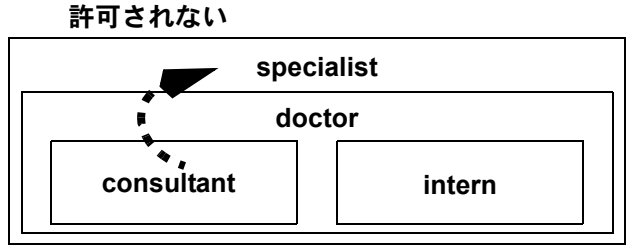
図 5-1: 明示的および暗黙的に付与された権限



- ある役割をその役割に含まれている別の役割に付与することはできません。これによって、階層内の「ループ」が回避されます。

  たとえば、図 5-2 では、役割 “specialist” を役割 “consultant” に付与することはできません。“consultant” は既に “specialist” に含まれているためです。

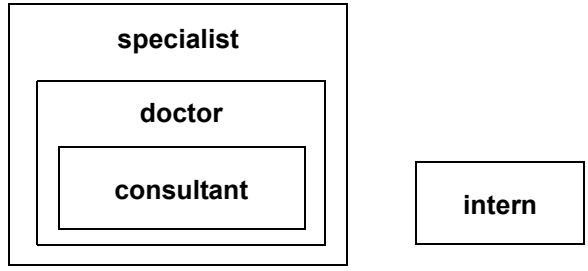
図 *5-2:* 付与者に含まれる役割に対する役割の付与



- システム・セキュリティ担当者がユーザに付与した役割に別の役割が含まれている場合は、そのユーザは、付与された役割に含まれるすべての役割におけるメンバシップを暗黙的に取得します。ただし、役割を直接アクティブ化または非アクティブ化できるのは、ユーザがその役割において明示的なメンバシップを持っている場合だけです。
- システム・セキュリティ担当者がある役割を別の役割に付与するとき、これらの役割がメンバシップ・レベルで明示的または暗黙的に相互排他である場合は、役割の付与はできません。

  たとえば、図 5-3 で、役割 “intern” が役割 “consultant” とメンバシップ・レベルで相互排他であると定義されている場合は、システム・セキュリティ担当者が “intern” を “doctor” に付与することはできません。

図 *5-3:* メンバシップでの相互排他性



- ユーザは直接付与された役割だけをアクティブ化、または非アクティブ化できます。

  たとえば、図 5-3 の階層で、役割 “specialist” がユーザに付与されたとします。ユーザには役割 “specialist” のすべてのパーミッションが付与されます。また、階層化されているので、役割 “doctor” と “consultant” のすべてのパーミッションも暗黙的に付与されます。ただし、このユーザがアクティブ化できるのは、役割 “specialist” だけです。“doctor” と “consultant” は、直接付与されたものではないので、アクティブ化はできません。「役割のアクティブ化と非アクティブ化」(157 ページ) を参照してください。

役割を他の役割から取り消す方法は、役割を他の役割に付与する方法に似ています。これによって包含関係が削除されますが、包含関係は直接的なものでなければなりません。

次に例を示します。

- システム・セキュリティ担当者が役割 “specialist” から役割 “doctor” を取り消すと、“specialist” には役割 “consultant” も “intern” も含まれなくなります。
- 役割 “specialist” から役割 “intern” を取り消すことはできません。これは “intern” が “specialist” に直接含まれるものではないからです。

ログイン時のデフォルト・アクティブ化の設定

システム・セキュリティ担当者は、alter login または alter login profile を使用して役割のアクティブ化を変更することができます。

ユーザが Adaptive Server にログインしたとき、デフォルトに設定された役割によっては、そのユーザの役割は必ずしもアクティブになりません。役割にパスワードが対応付けられている場合、ユーザは、set role コマンドを使用して、その役割をアクティブ化する必要があります。

システム・セキュリティ担当者はログイン時にデフォルトで付与された役割をアクティブ化するかどうかを決定し、alter login profile または alter login の auto activated roles 属性を使用して各ユーザのユーザ役割のデフォルト・ステータスを個々に設定します。個々のユーザが変更できるのは、自分自身のデフォルト設定だけです。auto activated roles はユーザ定義の役割だけに影響し、システム標準の役割には影響しません。

デフォルトでは、付与されたユーザ定義の役割はログイン時にアクティブ化されませんが、付与されたシステム標準の役割は、パスワードが対応付けられていなければ、自動的にアクティブ化されます。

次の例では、自動アクティブ化役割がパスワード保護されていない場合に、役割をログイン時に自動的にアクティブ化する方法を示します。

```
alter login mgr add auto activated roles
mgr_role, eng_role
```

次の例では、自動アクティブ化役割がパスワード保護されていない場合に、ログイン・プロファイルを使用して役割を自動的にアクティブ化する方法を示します。mgr_role と eng_role を mgr_lp に付与する必要があります。

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

ユーザ定義の役割の削除

システム・セキュリティ担当者として、次を使用して役割を削除します。

drop role *role_name* [with override]

ここで、*role_name* はユーザ定義の役割の名前です。

with override を指定すると、サーバ上のすべてのデータベースで、その役割に付与されているアクセス権限がすべて取り消されます。

override オプションを使用しない場合は、すべてのデータベースでその役割に付与された権限をすべて取り消してから、役割を削除してください。この処理を行わないと、コマンドは失敗します。権限を取り消すには、revoke コマンドを使用します。

役割を削除する前に、メンバシップを削除する必要はありません。役割を削除すると、with override オプションを使用するかどうかにかかわらず、その役割内のユーザ・メンバシップは自動的に削除されます。

役割のアクティブ化と非アクティブ化

役割をアクティブ化しなければ、アクセス権は得られません (つまり、アクティブではない役割には権限がありません)。デフォルトの役割はログイン時にアクティブにできません。パスワードが指定されている役割は、ログイン時は必ず非アクティブです。

役割をアクティブ化または非アクティブ化するには、次の構文を使用します。

set role *role_name* [with passwd "*password*"] {on | off}

役割をアクティブ化する場合のみ、with passwd パラメータを含めます。『リファレンス・マニュアル：コマンド』を参照してください。

たとえば、パスワード “sailing19” が設定されている役割 “financial_analyst” をアクティブ化するには、次のように入力します。

```
set role financial_analyst with passwd "sailing19" on
```

役割は必要なときにだけアクティブ化し、不要になったら非アクティブ化するようにしてください。sa_role がアクティブな場合は、使用するすべてのデータベース内でデータベース所有者として作業することに注意してください。

役割に関する情報の表示

表 5-2 は、役割に関する情報の表示に使用するシステム・プロシージャと関数を示します。

表 5-2: 役割について参照する情報

表示する情報	使用	参照箇所
役割名の役割 ID	role_id システム関数	「役割 ID と役割名の表示」(158 ページ)
役割 ID の役割名	role_name システム関数	「役割 ID と役割名の表示」(158 ページ)
システム標準の役割	show_role システム関数	「アクティブなシステム標準の役割の表示」(159 ページ)
役割階層、およびユーザに付与された役割	sp_displayroles システム・プロシージャ	「役割の階層の表示」(159 ページ)
役割階層内で、ある役割に他の役割が含まれているかどうか	role_contain システム関数	「階層内のユーザ定義の役割の表示」(159 ページ)
2 つの役割が相互排他的かどうか	mut_excl_roles システム関数	「相互排他性の判別」(159 ページ)
現在のセッションに対してアクティブな役割	sp_activeroles システム・プロシージャ	「役割のアクティブ化の判別」(160 ページ)
プロシージャを実行するために正しい役割がアクティブ化されているかどうか	has_role システム関数	「ストアド・プロシージャ内の役割の検査」(160 ページ)
ログイン (付与された役割を含む)	sp_displaylogin システム・プロシージャ	「ログイン・アカウントに関する情報の取得」(74 ページ)
ユーザ、グループ、または役割についてのパーミッション	sp_helprotect システム・プロシージャ	「パーミッションを表示する方法」(197 ページ)

役割 ID と役割名の表示

役割の名前がわかっている場合に、その役割 ID を表示するには、次の構文を使用します。

role_id(*role_name*)

すべてのユーザが role_id を実行できます。役割が有効ならば、role_id は、サーバワイドでのその役割の ID (srid) を返します。syssrvroles システム・テーブルの srid カラムに役割 ID が格納され、name カラムに役割名が格納されています。役割が無効な場合、role_id は NULL を返します。

役割 ID がわかっている場合に、その役割名を表示するには、role_name を使用します。

role_name(*role_id*)

すべてのユーザが role_name を実行できます。

アクティブなシステム標準の役割の表示

指定したログインの現在アクティブなシステム標準の役割を表示するには、show_role を使用します。

```
show_role()
```

ログインに対してシステム標準の役割が 1 つもアクティブ化されていない場合、show_role は NULL を返します。実行するユーザがデータベース所有者であり、別のユーザになり代わるために setuser を実行した後で show_role を実行した場合は、show_role はなり代わる別のユーザのアクティブなシステム標準の役割ではなく、そのユーザ自身のシステム標準の役割を返します。

すべてのユーザがシステム関数 show_role を実行できます。

注意　システム関数 show_role を実行しても、ユーザ定義の役割についての情報は表示されません。

役割の階層の表示

sp_displayroles を使用すると、ログイン名に付与されたすべての役割を表示することや、役割の階層ツリー全体をテーブル形式で表示することができます。

```
sp_displayroles {login_name | rolename [, expand_up | expand_down]}
```

すべてのユーザが、sp_displayroles を実行して各自の役割を表示できます。他のユーザに付与された役割に関する情報を表示できるのは、システム・セキュリティ担当者だけです。

階層内のユーザ定義の役割の表示

指定した役割に、指定した別の役割が含まれているかどうかを調べるには、role_contain を使用します。

```
role_contain (["role1", "role2"])
```

role1 が *role2* に含まれている場合、role_contain は 1 を返します。

すべてのユーザが role_contain 関数を実行できます。

相互排他性の判別

ユーザに割り当てられた 2 つの役割が相互排他の関係にあるかどうかと、その役割がどのレベルで相互排他であるかを調べるには、mut_excl_roles 関数を使用します。

```
mut_excl_roles(role1, role2, {membership | activation})
```

すべてのユーザが mut_excl_roles 関数を実行できます。指定した役割、または指定した役割に含まれる役割が相互排他の関係にある場合、mut_excl_roles は 1 を返します。役割が相互排他の関係にない場合、mut_excl_roles は 0 を返します。

役割のアクティブ化の判別

Adaptive Server の現在のログイン・セッションでアクティブな役割をすべて表示するには、次のコマンドを使用します。

```
sp_activeroles [expand_down]
```

expand_down を指定すると、ユーザに付与された役割に含まれるすべての役割の階層が表示されます。

すべてのユーザが sp_activeroles を実行できます。

ストアド・プロシージャ内の役割の検査

特定の役割を持つユーザだけがストアド・プロシージャを実行できることを保証するには、そのプロシージャの中で has_role を使用します。特定のストアド・プロシージャに対する不正なアクセスを防止して安全を保証するメカニズムは、has_role だけです。

grant execute を使用すると、ストアド・プロシージャに対する実行パーミッションを、指定の役割が付与されているすべてのユーザに付与できます。同様に、revoke execute を使用すると、このパーミッションを削除できます。

ただし、grant execute では、指定の役割を持たないユーザにストアド・プロシージャの実行パーミッションが付与されることを防ぐことはできません。たとえば、システム管理者以外のユーザに、ストアド・プロシージャを実行するパーミッションが決して付与されないようにするには、そのストアド・プロシージャの中で has_role を使用し、呼び出しを行うユーザに正しい役割があるかどうかを検査します。

has_role は、必要な役割の文字列を受け取り、呼び出し元がその役割を所有していれば 1 を返します。所有していなければ、0 を返します。

たとえば、次のプロシージャは、has_role を使用して、ユーザが役割 sa_role を持つかどうかを確認しています。

```
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)
```

役割の付与と取り消し

定義した役割は、サーバ内の任意のログイン・アカウントまたは役割に付与できます。ただし、相互排他性と階層の規則に違反しない場合に限ります。表 5-3 は、役割に関連するタスクとそのタスクの実行に必要な役割、および使用するコマンドを示します。

表 5-3: タスク、必要な役割、および使用するコマンド

タスク	必要な役割	コマンド
sa_role 役割の付与	システム管理者	grant role
sso_role 役割の付与	システム・セキュリティ担当者	grant role
oper_role 役割の付与	システム・セキュリティ担当者	grant role
ユーザ定義の役割の付与	システム・セキュリティ担当者	grant role
役割階層の作成	システム・セキュリティ担当者	grant role
役割階層の変更	システム・セキュリティ担当者	revoke role
システム標準の役割の取り消し	システム・セキュリティ担当者	revoke role
ユーザ定義の役割の取り消し	システム・セキュリティ担当者	revoke role

役割の付与

ユーザまたは他の役割に役割を付与するには、次の構文を使用します。

```
grant role role_granted [{, role_granted}...]
    to grantee [{, grantee}...]
```

各パラメータの意味は、次のとおりです。

- *role_granted* は、付与する役割です。付与する役割は、いくつでも指定できます。
- *grantee* は、ユーザ、役割、またはログイン・プロファイルの名前です。付与対象のユーザや役割はいくつでも指定できます。

grant 文で指定したすべての役割が、指定したすべてのユーザと役割に付与されます。ある役割を別の役割に付与すると、役割の階層が作成されます。

たとえば、Susan、Mary、John に役割 “financial_analyst” と “payroll_specialist” を付与するには、次のように入力します。

```
grant role financial_analyst, payroll_specialist
  to susan, mary, john
```

grant と役割について

grant コマンドを使用すると、システム標準の役割かユーザ定義の役割かに関係なく、指定した役割を付与されているすべてのユーザにオブジェクトのパーミッションを付与できます。これによって、次に示す役割を付与されているユーザに対してオブジェクトの使用を制限できます。

- システム標準の役割
- ユーザ定義の役割

役割は、ログイン・アカウント、別の役割、またはログイン・プロファイルに対してのみ付与できます。

役割にパーミッションを付与した場合、直接またはグループから、特定の役割を持たないユーザに同じパーミッションが付与されないようにすることはできません。たとえば、システム管理者のみが適切にストアド・プロシージャを実行できるようにするには、ストアド・プロシージャで has_role システム関数を使用して、ユーザが必須の役割を付与されていてアクティブ化していることを確認します。「役割に関する情報の表示」(158 ページ) を参照してください。

役割に付与されているパーミッションは、ユーザやグループに付与されているパーミッションよりも優先されます。たとえば、John がシステム・セキュリティ担当者の役割を付与されていて、sales テーブルに対するパーミッションが sso_role に付与されているとします。sales に対する John 個人のパーミッションが取り消されても、役割に付与されているパーミッションが個人に付与されているパーミッションよりも優先されるので、John は、sso_role をアクティブにすれば sales にアクセスできます。

役割の取り消し

revoke role を使用すると、ユーザ、その他の役割、およびログイン・プロファイルの役割を取り消すことができます。

```
revoke role role_name [{, role_name}...]from grantee [{, grantee}...]
```

各パラメータの意味は、次のとおりです。

- *role_name* は、取り消す役割の名前です。取り消す役割は、いくつでも指定できます。
- *grantee* は、ユーザまたは役割の名前です。付与対象のユーザや役割はいくつでも指定できます。

revoke 文で指定したすべての役割が、指定したすべてのユーザと役割から取り消されます。

ログイン・プロファイルに付与される役割

ログイン・プロファイルに付与される役割は、該当のプロファイルに割り当てられているユーザであれば誰でもアクティブ化できます。「ログイン・プロファイルへの役割の付与」(60 ページ) を参照してください。

役割のパスワードのセキュリティ保護

15.7 よりも前のバージョンの Adaptive Server では、役割のパスワードは Sybase 独自の暗号化を使用して syssrvroles システム・テーブル内に格納されていましたが、Adaptive Server バージョン 15.7 では、SHA-256 ダイジェストとして安全にディスク上に格納されます。

Adaptive Server をバージョン 15.7 以降にアップグレードし、アップグレード後に初めて役割のパスワードをアクティブ化する際、Adaptive Server はその役割パスワードを暗号化して SHA-256 ダイジェストとして格納します。

SHA-256 で暗号化された役割のパスワードをダウングレードすることはできません。その代わり、Adaptive Server はダウングレード時に役割パスワードをトランケートして役割をロックします。その後、管理者はダウングレード後にパスワードをリセットして役割のロックを解除する必要があります。

注意 高可用性環境では、プライマリ・サーバで初めて使用する際にアップグレードされる役割パスワードは、コンパニオン・サーバでもアップグレードされます。

文字セットについて

15.7 よりも前の Adaptive Server のバージョンでは、パスワードにサーバのデフォルトの文字セットを使用し、それを暗号化していました。これが変更され、現在はパスワードが自動的に標準の形式に、つまり、汎用の標準化された形式に変換されます。このように自動変換されることで、デフォルトの文字セットを変更する際に文字のマッピングが一致せずに役割のアクティブ化に失敗するという事態が回避されます。

役割のロックと *syssrvroles*

max failed_logins オプションを使用すると、役割のアクティブ化を試みて特定の回数失敗した後に役割を自動的にロックできます。手動の場合は、alter role *rolename* lock を使用します。Adaptive Server は、ロックされた役割についての情報を syssrvroles システム・テーブルに保存します。

- lockdate － 役割がロックされた日付を示します。lockdate は、役割がロックされた datetime に設定されます。
- locksuid － 役割のロックを実行した人物を示します。
- lockreason － 役割がロックされた理由を示します。lockreason は、国際化されたメッセージで表現できる整数にコード化されます。MSGDB データベースではそれぞれの理由にメッセージが追加されており、ローカル言語で理由を特定できます。

役割のロックが解除されていると、これらの値が NULL にリセットされます。

次に、その値と説明を示します。

lockreason の値	locksuid の値	役割の lockreason の説明
NULL	NULL	役割はロックされていません。
1	alter role の呼び出し元の suid	alter role *rolename* lock を手動で実行することにより suid によって役割がロックされます。
2	役割がロックされる原因となった役割のアクティブ化を最後に試行したユーザの suid	役割アクティブ化の失敗回数がログイン失敗最大回数に達したため、Adaptive Server によって役割がロックされました。

注意 高可用性機能を使用している場合、syssrvroles カラムを更新すると、プライマリ・サーバとコンパニオン・サーバの両方が更新されます。

役割パスワードを確認するログイン・パスワード・ポリシー

Adaptive Server バージョン 15.7 では、ログイン・パスワードに適用されるパスワード複雑性オプションが役割パスワードにも適用されます。役割パスワードにも適用されるオプションは、次のオプションで確認します。

- 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

パスワード・ポリシー・オプションに対する高可用性サポート

Adaptive Server の高可用性機能によって、プライマリ・サーバおよびセカンダリ・サーバ間のパスワード・ポリシー・オプションが同期化されます。

- 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 では、“password policy” 定属性を使用して、プライマリ・サーバとセカンダリ・サーバの両方の値の不整合性を確認します。高可用性アドバイザリ・チェックは、両方のサーバでこれらの値が同じであれば成功し、異なっていれば失敗します。次に例を示します。

```
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
```

出力の advisory カラムの値が 2 になっていますが、これは両サーバの値が一致していないので、ユーザがクラスタ・オペレーションを進めることができないことを示します。

sp_companion do_advisory の出力も、両方のサーバでの特定なパスワード・ポリシー・チェックの不整合性を示します。

役割のための Adaptive Server の設定

インストール

役割機能を使用する前に、Adaptive Server の master データベースに追加のディスク領域があり、syssrvroles テーブルに追加されるカラムのためのトランザクション・ログがあることも確認してください。データベース管理者は、alter database コマンドを使用すると領域を追加できます。

ページ当たりの役割の密度と役割パスワードの変更のために必要なログ領域を特定するには、sp_help syssrvroles と sp_helpdb を使用します。続いて、次の値を比較できます。

- パスワードを特定回数変更する前と後のログ領域の値
- 日付で syssrvroles を更新する set role with passwd コマンドの特定の数値

Adaptive Server のアップグレード

アップグレード・プロセス時、Adaptive Server は locksuid、lockreason、および lockdate を自動的に syssrvroles に追加します。これらのカラムでは NULL が許可され、アップグレード後にデフォルト値の NULL が入ります。値は必要な場合にのみ設定されます。

Adaptive Server のダウングレード

Adaptive Server をバージョン 15.5 にダウングレードすると、Adaptive Server は役割パスワードをトランケートしてロックします。また、Adaptive Server では役割パスワードについて allow password downgrade の使用がサポートされません。

ダウングレード後、役割パスワードと役割アカウントを再度使用するには、管理者が役割パスワードをリセットして役割アカウントのロックを解除する必要があります。

ダウングレード・プロセス時に、Adaptive Server は次の動作を実行します。

- 役割パスワードのトランケートと役割のロック
- クラス 35 内の sysattributes にある属性とクラス 35 そのものの削除

- syssrvroles からの locksuid カラム、lockreason カラム、および lockdate カラムの削除

パスワードのダウングレードは、シングルユーザ・モードで sp_downgrade を実行して行います。“-m” コマンド・ライン・オプションで始まる dataserver を使用すると、サーバがシングルユーザ・モードで起動し、システム管理者のみがログインできるようになります。

次の例では、sp_downgrade を実行することで “doctor_role” 役割のパスワードがロックおよびトランケートされます。管理者は、これらの役割のパスワードをリセットできるように、個の出力をファイルにリダイレクトできます。

```
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)
```

上記のダウングレード手順の例のエラー・ログ出力の例にあるように、sp_downgrade 実行時の手順や発生する可能性があるシステム・エラーを特定するために、追加のメッセージがエラー・ログに表示されます。

```
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.
```

sp_downgrade を実行した後で、データやシステム・カタログを変更する可能性のある新しいログインやその他のアクションを避けるには、サーバをシャットダウンします。

Adaptive Server をバージョン 15.7 で再起動する場合は、次の点に注意してください。

- sp_downgrade が正常に実行されて、サーバがシャットダウンされると、Adaptive Server では内部アップグレード・アクションが再び実行されて、システム・テーブルの変更がバージョン 15.7 にアップグレードされます。
- バージョンを戻す予定の Adaptive Server の旧バージョンを起動する前に、再度 sp_downgrade を実行する必要があります。

ロックされている役割とトランケートされたパスワードを有効にすることができます。次の例では、ダウングレード・プロセスによって “doctor_role” がロックされてそのパスワードがトランケートされることが sp_displayroles の出力に示されています。

```
select srid,status,name,password from syssrvroles
go
suid   status  name                password
------ ------- ------------------- -----
33     2       doctor_role         NULL
```

役割のロックを解除します。

```
alter role doctor_role unlock
```

役割の新しいパスワードを設定します。

```
alter role doctor_role add passwd "dProle1"
```

ここで sp_displayroles を実行すると、役割のロックが解除されてパスワードが設定されたことが表示されます。

```
select srid,status,name,
"vers"=substring(password,2,1) from syssrvroles
go
suid   status  name                vers
------ ------- ------------------- -----
33     0       doctor_role         0x05
```

第 6 章

ユーザ・パーミッションの管理

この章では、ユーザ・パーミッションの使用と実装について説明します。

概要

「任意アクセス制御 **(DAC)**」を使用すると、ユーザの ID、グループのメンバシップ、アクティブな役割に基づいて、オブジェクトやコマンドに対するアクセスを制限できます。オブジェクト所有者などの特定のアクセス・パーミッションを持つユーザは、そのアクセス・パーミッションを他のユーザに渡すかどうかを選択できるので、制御は「任意」であると言えます。

Adaptive Server の任意アクセス制御システムは、次のタイプのユーザを識別します。

- 1 つまたは複数のシステム定義の役割 (システム管理者、システム・セキュリティ担当者、オペレータ、およびその他の役割) を処理するユーザ
- データベース所有者
- データベース・オブジェクト所有者
- その他のユーザ

システム管理者 (sa_role を持つユーザ) は、DAC システムの外部で操作を行い、暗号化キー (『暗号化カラム・ユーザーズ・ガイド』を参照) を除くすべてのデータベース・オブジェクトに対するアクセス・パーミッションを常に所有しています。システム・セキュリティ担当者は、常に sybsecurity データベース内の監査証跡テーブルにアクセスしてシステム管理者によるアクセスを追跡できます。

sa_role を持つユーザの場合は、master データベースで grant コマンドを発行すると、all により create database、set tracing、および connect のパーミッションも同様に付与されます。

データベース所有者は、他のユーザが所有するオブジェクトに対するパーミッションを自動的に受け取るわけではありませんが、以下のことが実行できます。

- setuser コマンドを使用して、データベース内の特定のユーザの ID を持ち、そのユーザのすべてのパーミッションを一時的に取得する。
- setuser コマンドを使用してオブジェクト所有者の ID を持ち、次に grant コマンドを使用してパーミッションを付与することによって、そのオブジェクトに対するパーミッションを永続的に取得する。

別のユーザの ID を使用して、データベースまたはオブジェクトに対するそのユーザのパーミッションを取得する方法については、「別のユーザのパーミッションの取得」(187 ページ) を参照してください。

オブジェクト所有者は、オブジェクトへのアクセス権を他のユーザに付与したり、アクセス・パーミッションを他のユーザに渡す権限を他のユーザに付与したりすることもできます。grant コマンドを使用すると、ユーザ、グループ、役割に対して各種のパーミッションを与えることができます。また、revoke コマンドを使用するとパーミッションを無効にできます。grant と revoke コマンドを使用して、次のパーミッションをユーザに与えます。

- データベースの作成
- データベース内のオブジェクトの作成
- dbcc や set proxy などの特定のコマンドの実行
- 指定したテーブル、ビュー、ストアド・プロシージャ、暗号化キー、カラムへのアクセス

grant と revoke を使用して、システム・テーブルに対するパーミッションも設定できます。

デフォルトで “public” に与えられるパーミッションについては、grant 文や revoke 文の実行は不要です。

すべてのユーザがいつでもパーミッションなしで使用できるコマンドもあります。また、特定ステータスのユーザしか使用できず、譲渡できないコマンドもあります。

権限の付与や取り消しが可能なコマンドに対するパーミッションを割り当てることができるかどうかは、各ユーザの役割やステータス (システム管理者、データベース所有者、システム・セキュリティ担当者、データベース・オブジェクト所有者など) と、そのユーザが持つ役割に付与されているパーミッションにそのパーミッションを他のユーザに付与するオプションが付いているかどうかによって決まります。

ビューとストアド・プロシージャをセキュリティ・メカニズムとして使用することもできます。「セキュリティ・メカニズムとしてのビューとストアド・プロシージャの使用」(201 ページ) を参照してください。

データベース作成用のパーミッション

create database コマンドを使用するパーミッションを付与できるのは、システム管理者だけです。create database のパーミッションを受け取るユーザは、master データベースの有効なユーザでもある必要があります。これは、データベースの作成は master を使用している状態で行われるためです。

多くのインストール環境では、システム管理者だけが create database パーミッションを持ち、データベースの配置とデータベース・デバイスの領域の割り付けを集中管理します。このような状況では、システム管理者が他のユーザに代わって新しいデータベースを作成し、所有権を該当するユーザに譲渡します。

別のユーザに所有させるデータベースを作成するには、次の手順に従います。

1　master データベース内で create database コマンドを発行します。

2　use コマンドを使用して、作成した新しいデータベースに切り替えます。

3　sp_changedbowner を実行します。

データベース所有権の変更

sp_changedbowner を使うと、データベースの所有権を変更できます。システム管理者はユーザ・データベースを作成して、必要な初期設定作業を完了してからその所有権を別のユーザに付与できます。sp_changedbowner を実行できるのはシステム管理者だけです。

ユーザをデータベースに追加する前、およびそのユーザによってデータベース内にオブジェクトが作成される前に所有権を譲渡することをおすすめします。新しい所有者は Adaptive Server 上に既にログイン名を持っている必要がありますが、そのデータベースのユーザであったりデータベースにエイリアスを持っていたりしてはなりません。そのような場合は、sp_dropuser または sp_dropalias を実行してからでなければ、データベースの所有権は変更できません。また、ユーザを削除する前に、オブジェクトの削除が必要なこともあります。

sp_changedbowner は、所有権を変更するデータベース内で発行します。構文は次のとおりです。

sp_changedbowner *loginame* [, true]

ユーザ “albert” を現在のデータベースの所有者にして、元の “dbo” ユーザのエイリアスを削除する例を次に示します。

```
sp_changedbowner albert
```

エイリアスとそのパーミッションを新しい “dbo” に移動するには、true パラメータを指定します。

注意 master、model、tempdb、または sybsystemprocs のデータベースの所有権は変更できません。その他のシステム・データベースの所有権も変更しないでください。

データベース所有者の権限

オブジェクト作成パーミッションを他のユーザに付与できるのは、データベース所有者とシステム管理者だけです (ただし、**暗号化キー作成**および**トリガ作成**パーミッションを付与できるのは、システム・セキュリティ担当者だけです)。データベース所有者は、そのデータベース内であらゆる作業を実行する権限を持っています。また、grant コマンドを使って他のユーザに明示的にパーミッションを付与しなければなりません。

次のコマンドを使用するためのパーミッションは、自動的にデータベース所有者に付与され、他のユーザに渡すことはできません。

- checkpoint
- dbcc
- alter database
- online database
- drop database
- dump database

- dump transaction
- grant (オブジェクト作成パーミッション)
- load database
- load transaction
- revoke (オブジェクト作成パーミッション)
- setuser

データベース所有者は、次のようにパーミッションの付与と取り消しを行うことができます。

- create default、create procedure、create rule、create table、create view の各コマンドの使用。

  データベース所有者は、sa_role を持ち、master データベースを使用している場合、create database、set tracing、および connect を使用するためのパーミッションを付与できます。

- all − データベース所有者である場合、all を実行すると、create database、create trigger、および create encryption key 以外のすべての create コマンドのパーミッションが付与されます。
- default permissions on system tables
- dbcc コマンドの使用：checkalloc、checkcatalog、checkdb、checkindex、checkstorage、checktable、checkverify、fix_text、indexalloc、reindex、tablealloc、textalloc、tune。

データベース・オブジェクト所有者

データベース・オブジェクト (テーブル、ビュー、暗号化キー、またはストアド・プロシージャ) を作成するユーザはそのオブジェクトの所有者となり、そのオブジェクトに対するすべてのオブジェクト・アクセス・パーミッションを自動的に付与されます。オブジェクト所有者以外のユーザ (データベースの所有者も含む) は、オブジェクト所有者またはそのオブジェクトに対するパーミッションを付与する grant パーミッションを持つユーザによって明示的にパーミッションを付与されない限り、そのオブジェクトに対するすべてのパーミッションを自動的に拒否されます。

たとえば、Mary が pubs2 データベースの所有者であり、そのデータベース内にテーブルを作成するためのパーミッションを Joe に付与したとします。Joe は、テーブル new_authors を作成し、このデータベース・オブジェクトの所有者になります。

初めは、new_authors のオブジェクト・アクセス・パーミッションを持つのは Joe だけです。Joe は、このテーブルに対するオブジェクト・アクセス・パーミッションを他のユーザに付与したり取り消したりできます。

次のオブジェクト変更パーミッションは、デフォルトではテーブルの所有者にあり、他のユーザに譲渡することはできません。

- alter table
- drop table
- create index

特定のデータベース・オブジェクトに対する select、insert、update、delete、references、decrypt、truncate table、update statistics、delete statistics、execute の各パーミッションを特定のユーザに付与する grant コマンドと revoke コマンドを使用するためのパーミッションは、grant with grant option コマンドを使って譲渡することができます。

オブジェクト (テーブル、ビュー、インデックス、ストアド・プロシージャ、ルール、暗号化キー、トリガ、またはデフォルト) を削除するための drop パーミッションは、デフォルトではオブジェクト所有者にあり、他のユーザには譲渡できません。

その他のデータベース・ユーザの権限

その他のデータベース・ユーザへのパーミッションの付与と取り消しは、オブジェクト所有者、データベース所有者、grant オプションでパーミッションを付与されたユーザ、システム管理者、またはシステム・セキュリティ担当者が行います。これらのユーザはユーザ名、グループ名、またはキーワード public によって指定します。

役割をアクティブ化すると、ユーザに割り当てられている役割に付与されたパーミッションをすべてのユーザーが継承します。

システム・プロシージャに対するパーミッション

システム・プロシージャに対するパーミッションは、システム・プロシージャが格納されている sybsystemprocs データベースで設定します。

セキュリティ関連のシステム・プロシージャを実行できるのは、システム・セキュリティ担当者だけです。その他のシステム・プロシージャの中には、システム管理者しか実行できないものもあります。

また、データベース所有者しか実行できないシステム・プロシージャもあります。これらのプロシージャは、プロシージャを実行するユーザが、プロシージャの実行元であるデータベースの所有者であることを確認します。

その他のシステム・プロシージャは、パーミッションを付与されているユーザであれば実行できます。つまり、ユーザはシステム・プロシージャを実行するためのパーミッションをすべてのデータベースで持つか、あるいは、どのデータベースでも持たないかのどちらかです。

sybsystemprocs..sysusers に登録されていないユーザは、sybsystemprocs では “guest” として扱われ、多くのシステム・プロシージャに対するパーミッションを自動的に付与されます。システム・プロシージャに対するユーザのパーミッションを取り消すには、システム管理者がそのユーザを sybsystemprocs..sysusers に追加して、そのプロシージャに適用される revoke 文を発行する必要があります。ユーザ・データベースの所有者が自分のデータベースからシステム・プロシージャに対するパーミッションを直接制御することはできません。

パーミッションの付与と取り消し

grant と revoke を使用して、次のタイプのパーミッションを制御できます。

- オブジェクト・アクセス・パーミッション
- 関数から選択するパーミッション
- コマンドを実行するパーミッション
- dbcc コマンドを実行するパーミッション
- 一部の set コマンドを実行するパーミッション
- システム・テーブルのデフォルト・パーミッション

各データベースには、独自の独立した保護システムがあります。あるデータベースで特定のコマンドを使用するためのパーミッションを与えられても、そのユーザに、他のデータベースでそのコマンドを使用するためのパーミッションが与えられるわけではありません。

オブジェクト・アクセス・パーミッション

オブジェクト・アクセス・パーミッションは、特定のデータベース・オブジェクトにアクセスする特定のコマンドの使用を規制します。たとえば、authors テーブルに対して select コマンドを使用するには、パーミッションがユーザに明示的に付与されていなければなりません。オブジェクト・アクセス・パーミッションの付与と取り消しは、オブジェクト所有者 (およびシステム管理者またはシステム・セキュリティ担当者) が行います。オブジェクト所有者は、他のユーザにこのパーミッションを付与できます。

表 6-1 は、オブジェクト・アクセス・パーミッションのタイプと、そのパーミッションが適用されるオブジェクトを示します。

表 6-1: パーミッションと適用するオブジェクト

パーミッション	オブジェクト
select	テーブル、ビュー、カラム
update	テーブル、ビュー、カラム
insert	テーブル、ビュー
delete	テーブル、ビュー
references	テーブル、カラム
execute	ストアド・プロシージャ
truncate table	テーブル
delete statistics	テーブル
update statistics	テーブル
decrypt	テーブル、ビュー、カラム
select	暗号化キー

references パーミッションとは、alter table コマンドや create table コマンドで指定できる参照整合性制約のことです。decrypt パーミッションとは、暗号化カラムを復号化するために必要なパーミッションのことです。暗号化キーの select パーミッションとは、create table、alter table、または select into の各コマンドの暗号化キーを使用してカラムを暗号化するために必要なパーミッションのことです。それ以外のパーミッションは、SQL コマンドのことを指します。オブジェクト・アクセス・パーミッションは、デフォルトでは、オブジェクト所有者、システム管理者、または暗号化カラムの decrypt および暗号化キーの select に関するシステム・セキュリティ担当者にあり、他のユーザに付与できます。

あるオブジェクトへのアクセス権が、複数のユーザから特定のユーザに付与された場合、付与されたユーザのアクセス権は、付与したすべてのユーザがそのアクセス権を取り消すまで有効です。システム管理者によってアクセス権が取り消された場合は、別のユーザからそのユーザにアクセス権が付与されていても、そのユーザのアクセスは拒否されます。

オブジェクト・アクセス・パーミッションを付与するには、grant コマンドを使用します。詳細については、『リファレンス・マニュアル：コマンド』を参照してください。

具体的 ID

Adaptive Server は、セッション中のユーザをログイン名によって識別します。この識別は、サーバのすべてのデータベースで有効です。ユーザがオブジェクトを作成すると、所有者のデータベース・ユーザ ID (*uid*) と作成者のログイン名の両方が、sysobjects テーブル内でオブジェクトと関連付けられます。この情報によって、どのユーザが所有するオブジェクトであるかが具体的に識別されるため、サーバは、いつオブジェクトのパーミッションが暗黙的に許可できるかを認識できます。

Adaptive Server ユーザがテーブルを作成し、そのテーブルにアクセスするプロシージャを作成したとき、そのプロシージャを使用するパーミッションを付与されたユーザには、そのオブジェクトに直接アクセスするためのパーミッションは必要ありません。たとえば、次のように “mary” というユーザに proc1 に対するパーミッションを付与したとき、mary はテーブル table1 に対する選択パーミッションは明示的に与えられてはいませんが、このテーブルのカラム id と descr を参照できます。

```
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
```

ただし、オブジェクトを具体的に識別できる場合にのみ暗黙的パーミッションが有効となることもあります。たとえば、エイリアスとデータベース間オブジェクト・アクセスの両方が関係する場合です。

SQL92 標準に準拠するための要件

set コマンドを使用して ansi_permissions を on にした場合は、update 文と delete 文を実行するための追加のパーミッションが必要です。表 6-2 は、必要となるパーミッションをまとめたものです。

表 6-2: update と delete に必要な ANSI パーミッション

	必要なパーミッション：*set ansi_permissions off*	**必要なパーミッション：*set ansi_permissions on***
update	値を設定するカラムに対する update パーミッション	値を設定するカラムに対する update パーミッション および where 句に指定するすべてのカラムに対する select パーミッション set 句の右側のすべてのカラムに対する select パーミッション
delete	テーブルに対する delete パーミッション	ローを削除するテーブルに対する delete パーミッション および where 句に指定するすべてのカラムに対する select パーミッション

ansi_permissions が on の場合に、必要となる追加の select パーミッションが与えられていないユーザが更新または削除を行うと、トランザクションはロールバックされ、エラー・メッセージが表示されます。このエラー・メッセージが表示された場合は、すべての関係するカラムに対する select パーミッションをオブジェクト所有者が付与する必要があります。

オブジェクト・アクセス・パーミッションの付与の例

次の文は、titles テーブルに対して挿入と削除を行うためのパーミッションを Mary と “sales” グループに付与します。

```
grant insert, delete
on titles
to mary, sales
```

次の文はストアド・プロシージャ makelist を使用するためのパーミッションを Harold に付与します。

```
grant execute
on makelist
to harold
```

次の文は、カスタム・ストアド・プロシージャ sa_only_proc を実行するためのパーミッションを、システム管理者の役割を付与されているユーザに付与します。

```
grant execute
on sa_only_proc
to sa_role
```

次の文は、authors テーブルに対して選択、更新、削除を行うためのパーミッションと、他のユーザに同じパーミッションを付与するためのパーミッションを Aubrey に付与します。

```
grant select, update, delete
on authors
to aubrey
with grant option
```

オブジェクト・アクセス・パーミッションの取り消しの例

次の 2 つの文はどちらも、titles テーブルの price カラムと total_sales カラムを更新するためのパーミッションをテーブル所有者以外のすべてのユーザから取り消します。

```
revoke update
on titles (price, total_sales)
from public
```

次の文は、authors テーブルを更新するためのパーミッションを Clare から取り消すと同時に、Clare がそのパーミッションを付与したすべてのユーザからもそのパーミッションを取り消します。

```
revoke update
on authors
from clare
cascade
```

次の文は、ストアド・プロシージャ new_sproc を実行するためのパーミッションをオペレータから取り消します。

```
revoke execute
on new_sproc
from oper_role
```

dbcc コマンドのパーミッションの付与

システム管理者は、Adaptive Server のシステム管理者レベルの権限を持たないユーザや役割に対して、dbcc コマンドを実行するパーミッションを付与できます。この「任意アクセス制御」により、システム管理者はデータベース・オブジェクトまたは特定のデータベース・レベルとサーバ・レベルのアクションへのアクセスを制御できます。

dbcc 構文の詳細については、『リファレンス・マニュアル：コマンド』を参照してください。

サーバワイドとデータベース固有の *dbcc* コマンド

dbcc コマンドは、次のいずれかです。

- データベース固有 － 特定のターゲット・データベースに対して実行する dbcc コマンド (checkalloc、checktable、checkindex、checkstorage、checkdb、checkcatalog、checkverify、fix_text、indexalloc、reindex、tablealloc、textalloc など)。これらのコマンドは特定のデータベースを対象としたコマンドですが、パーミッションの付与や取り消しができるのはシステム管理者だけです。
- サーバワイド － tune コマンドなど、サーバ全体に作用するが、特定のデータベースには関連付けられていない dbcc コマンド。これらのコマンドのパーミッションはデフォルトでサーバワイドに付与され、どのデータベースにも関連付けられません。

システム管理者は、これらのデータベース内で有効なユーザとして設定することで、すべてのデータベースで dbcc コマンドを実行するパーミッションをそのユーザに付与できます。ただし、grant dbcc コマンドのパーミッションをユーザに個別に付与すると、各ユーザを手動でデータベースに追加しなければなりません。パーミッションを役割に対して付与すれば、ユーザは “guest” ユーザとしてデータベースを使用できるようになるので、こちらの方法がより便利です。

セキュリティ管理の観点から、データベース固有の dbcc コマンドのパーミッションをサーバワイドに付与する方法をシステム管理者が選ぶこともあります。たとえば、すべてのデータベースに対する grant dbcc checkstorage を storage_admin_role というユーザ定義の役割に対して実行すれば、storage_admin_role に対する grant dbcc checkstorage をデータベースごとに実行する手間が省けます。

次のコマンドは、サーバワイドで有効なコマンドですが、データベース固有のコマンドではありません。

- tune などのサーバワイド dbcc コマンド
- storage_admin_role に対して付与される grant dbcc checkstorage など、サーバワイドにパーミッションが付与されるデータベース固有の dbcc コマンド

dbcc コマンドのパーミッションの付与対象者とデータベース内のユーザ

grant dbcc コマンドと revoke dbcc コマンドは、データベース内のユーザに対して機能します。

データベース内の役割に対して初めて grant が実行されると、その役割は自動的にユーザとして追加されるため、役割に dbcc の権限を付与するための追加の要件はありません。ログインは、パーミッションが付与されるデータベース内の有効なユーザでなければなりません。有効なユーザには “guest” が含まれます。

サーバワイドな dbcc コマンドの場合、ログインは master データベース内の有効なユーザでなければなりません。また、システム管理者はパーミッションの付与を master データベース内から実行する必要があります。

データベース固有の dbcc コマンドの場合、ログインはターゲット・データベース内の有効なユーザでなければなりません。

システム・テーブルのパーミッション

システム・テーブルで使うパーミッションは、他のテーブルのパーミッションと同じくデータベース所有者が制御できます。データベースを作成すると、一部のシステム・テーブルの select パーミッションが public に付与され、一部のシステム・テーブルの select パーミッションが管理者に制限されます。テーブルによっては、いくつかのカラムで、public に対する select パーミッションが制限されている場合もあります。

特定のシステム・テーブルに対する現在のパーミッションを調べるには、次のように実行します。

```
sp_helprotect system_table_name
```

たとえば、master データベースの syssrvroles のパーミッションを調べるには、次のコマンドを実行します。

```
use master
go
sp_helprotect syssrvroles
go
```

デフォルトでは、データベース所有者も含め、ユーザがシステム・テーブルを直接変更することはできません。代わりに、T-SQL コマンドと Adaptive Server に付属するシステム・プロシージャを使用してシステム・テーブルを変更します。これは整合性の保証に役立ちます。

警告！ Adaptive Server にはシステム・テーブルを変更できるメカニズムがありますが、システム・テーブルの変更はしないことを強くおすすめします。

システム・テーブルとストアド・プロシージャへのデフォルト・パーミッションの付与

grant コマンドと revoke コマンドでは、default permissions パラメータを指定できます。installmodel または installmaster では、システム・テーブル (次の表を参照) のデフォルト・パーミッションは付与されません。代わりに、Adaptive Server が新しいデータベースを構築するときに、これらのシステム・テーブルのデフォルト・パーミッションが割り当てられます。構文の一部は次のとおりです。

```
grant default permissions on system tables
revoke default permissions on system tables
```

default permissions on system tables は、任意のデータベースからこのコマンドを発行するときに、次のシステム・テーブルのデフォルト・パーミッションの付与または取り消しを指定します。

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	

デフォルトのパーミッションでは select がすべてのシステム・テーブルの public に適用されますが、次の例外があります。

- public から syscolumns(encrkeyid) の select を取り消す。
- public から syscolumns(encrkeydb) の select を取り消す。
- sso_role に syscolumns の select を付与する。
- public から sysobjects(audflags) パーミッションを取り消す。

- sysobjects のパーミッションを sso_role に付与する。
- public から sysencryptkeys のすべてのカラムに対する select を取り消す。
- sysencryptkeys のすべてのカラムに対する select を sso_role に付与する。

このコマンドを master データベースから実行すると、次のシステム・テーブルのデフォルト・パーミッションが付与または取り消されます。

syscharsets	syslanguages	sysmessages	sysservers
sysconfigures	syslisteners	sysmonitor	syssessions
syscurconfigs	syslocks	sysprocesses	syssrvroles
sysdatabases	syslogin	sysremotelogins	systimeranges
sysdevices	sysloginrole	sysresourcelimits	systransactions
sysengines	syslogshold	syssecmechs	sysusages

このコマンドでは次の変更も行われています。

- public から sysdatabases(audflags) の select を取り消す。
- public から syscolumns(encrkeyid) の select を取り消す。
- public から syscolumns(encrkeydb) の select を取り消す。
- sso_role に syscolumns の select を付与する。
- public から sysdatabases(deftabaud) の select を取り消す。
- public から sysdatabases(defvwaud) の select を取り消す。
- public から sysdatabases(defpraud) の select を取り消す。
- public から sysdatabases(audflags2) の select を取り消す。
- sysdatabases に対する select を sso_role に付与する。
- public から syslogins(password) の select を取り消す。
- public から syslogins(audflags) の select を取り消す。
- sso_role に syslogins の select を付与する。
- public から syslisteners(net_type) の select を取り消す。
- public から syslisteners(address_info) の select を取り消す。
- syslisteners に対する select を sso_role に付与する。
- public から syssrvroles(srid) の select を取り消す。
- public から syssrvroles(name) の select を取り消す。
- public から syssrvroles(password) の select を取り消す。
- public から syssrvroles(pwdate) の select を取り消す。
- public から syssrvroles(status) の select を取り消す。

- public から syssrvroles(logincount) の select を取り消す。
- syssrvroles に対する select を sso_role に付与する。
- public から sysloginroles(suid) の select を取り消す。
- public から sysloginroles(srid) の select を取り消す。
- public から sysloginroles(status) の select を取り消す。
- sso_role から sysloginroles に対する select を取り消す。

grant 文と *revoke* 文の組み合わせ

特定のパーミッションを特定のユーザに割り当てることができますが、ほとんどのユーザにほとんどの権限を付与するのであれば、すべてのユーザにすべてのパーミッションを付与してから、特定のユーザから特定のパーミッションを取り消す方が簡単です。

たとえば、データベース所有者は次の文を発行することによって、titles テーブルに対するすべてのパーミッションをすべてのユーザに付与できます。

```
grant all
on titles
to public
```

次に、次のような一連の revoke 文を発行します。

```
revoke update
on titles (price, advance)
from public
revoke delete
on titles
from mary, sales, john
```

grant 文と revoke 文の結果は、実行する順序によって異なります。競合を発生した場合は、後で発行された方の文が有効になります。

注意 SQL の規則では、grant コマンドは revoke コマンドよりも前に使用する必要がありますが、この 2 つのコマンドを同じトランザクション内で使用することはできません。したがって、オブジェクトへのアクセス権を “public” に付与した後で個別のユーザからそのアクセス権を取り消したとしても、そのユーザがこのオブジェクトにアクセスできる期間が、短期間ではあっても生じてしまいます。これを避けるには、create schema コマンドを使用して、1 つのトランザクション内に grant 句と revoke 句を指定してください。

パーミッションの順序と階層について

grant 文と revoke 文は発行順序が重要です。たとえば、titles テーブルに対する select パーミッションが Jose のグループに付与された後で、advance カラムを選択するための Jose のパーミッションが取り消された場合に、Jose が選択できるのは advance 以外のすべてのカラムですが、Jose と同じグループの他のユーザはこの場合もすべてのカラムを選択できます。

グループまたは役割に適用される grant 文や revoke 文は、そのグループまたは役割のメンバに割り当てられている競合するパーミッションを変更します。たとえば、titles テーブルの所有者が sales グループのメンバごとに異なるパーミッションを付与した後で、sales グループのメンバ全員に同じパーミッションを付与することにしたとします。その所有者は次の文を発行します。

```
revoke all on titles from sales
grant select on titles(title, title_id, type,
        pub_id)
    to sales
```

同じように、public に対して発行された grant 文と revoke 文は、以前に発行されたパーミッションの中で新しい状況と競合するすべてのパーミッションを、すべてのユーザについて変更します。

同じ grant 文と revoke 文でも、発行順序が異なると、結果もまったく異なります。たとえば、次の順序でこれらの文を発行すると、public グループに属する Jose は titles に対する select パーミッションを持たなくなります。

```
grant select on titles(title_id, title) to jose
revoke select on titles from public
```

これに対して、同じ文を逆の順序で発行すると、title_id と title カラムだけに対する select パーミッションを Jose だけが持つようになります。

```
revoke select on titles from public
grant select on titles(title_id, title) to jose
```

grant にキーワード public を使用した場合は、自分自身も含まれることを忘れないでください。オブジェクト作成パーミッションに対して revoke を実行するユーザは、データベース所有者でなければ public に含まれます。オブジェクト・アクセス・パーミッションに対して revoke を実行するユーザは、オブジェクト所有者でなければ public に含まれます。自分のテーブルを使用するための自分のパーミッションを取り消す一方で、そのテーブル上に作成されたビューにアクセスするためのパーミッションを自分自身に付与することもできます。このようにするには、grant 文と revoke 文を発行して明示的に自分のパーミッションを設定する必要があります。方針が変わった場合は、grant 文を使っていつでもパーミッションを再設定できます。

grant dbcc および set proxy の fipsflagger に対する警告の発行

set fipsflagger オプションが有効になっているときに grant dbcc と set proxy を発行すると、次の警告が発行されます。

```
行番号 %1! の SQL 文に ANSI 以外のテキストがあります。DBCC を使用したために、エラーが発生しました。
```

別のユーザのパーミッションの取得

Adaptive Server には、別のユーザの ID とパーミッション・ステータスを取得する方法が 2 つあります。

- データベース所有者は、setuser コマンドを使用して、現在のデータベース内の別のユーザになり代わり、その ID とパーミッション・ステータスを利用することができます。「setuser の使用」(187 ページ) を参照してください。
- 「代理権限」を利用すると、1 人のユーザがサーバ全体で別のユーザの ID を利用できます。詳細については、「代理権限の使用」(188 ページ) を参照してください。

setuser の使用

データベース所有者は、次の場合に setuser を使用できます。

- 別のユーザが所有するオブジェクトにアクセスする場合。
- 別のユーザが所有するオブジェクトに対するパーミッションを付与する場合。
- 別のユーザが所有者となるオブジェクトを作成する場合。
- 何らかの理由で別のユーザの DAC パーミッションを一時的に利用する場合。

setuser コマンドを実行すると、データベース所有者は自動的に別のユーザの DAC パーミッションを取得できますが、このコマンドは既に付与されている役割には影響しません。

setuser パーミッションは、デフォルトではデータベース所有者に付与されており、譲渡することはできません。なり代わるユーザは、そのデータベースのアクセス権を持つユーザでなければなりません。Adaptive Server は、なり代わるユーザのパーミッションをチェックします。

システム管理者は、setuser を使用して、別のユーザが所有するオブジェクトを作成できます。ただし、システム管理者は、DAC パーミッション・システムの外部で操作するため、setuser を使用して別のユーザのパーミッションを取得する必要はありません。setuser コマンドは、次の setuser コマンドが実行されるか、現在のデータベースが変更されるか、あるいはユーザがログオフするまで有効です。

構文は次のとおりです。

```
setuser ["user_name"]
```

この *user_name* は、ID を使用される、データベース内の有効なユーザです。

元の ID に戻るには、*user_name* の値を指定しないで setuser コマンドを実行します。

次の例は、データベース所有者が、Mary が所有する authors テーブルを読み込むパーミッションを Joe に付与する方法を示します。

```
setuser "mary"
grant select on authors to joe
setuser     /*reestablishes original identity*/
```

代理権限の使用

Adaptive Server の代理権限機能を使用すると、システム・セキュリティ担当者は、別のユーザのセキュリティ・コンテキストを利用する機能を、選択したログインに付与できます。また、さまざまなユーザに代わってアプリケーションでタスクを実行する方法を制御できます。代理権限を使用するパーミッションを持つログインは Adaptive Server 内の別のログインになり代わることができます。

警告！ 他のユーザ ID を利用する機能は非常に強力なものであるため、信頼された管理者とアプリケーションだけに利用を限定する必要があります。grant set proxy ... restrict role を使用すると、ID を切り替えたときにユーザが特定の役割を取得できないように制限できます。

set proxy または set session authorization を実行するユーザは、被代理ユーザのログイン名とサーバ・ユーザ ID の両方を使用して操作を行います。ログイン名は、master..syslogins の name カラムに保管されています。また、サーバ・ユーザ ID は、master..syslogins の suid カラムに保管されています。これらの値は、サーバ全体のすべてのデータベース内でアクティブです。

注意 set proxy と set session authorization の機能は同じなので、どちらを使用してもかまいません。唯一の違いは、set session authorization が ANSI SQL92 互換であるのに対し、set proxy は Transact-SQL の拡張機能であるという点です。

set proxy を使用した役割の制限

set proxy...restricted role を付与することによって、ID を切り替えたときに特定の役割を取得できないように制限できます。

set proxy の構文は次のとおりです。

```
grant set proxy to user | role
    [restrict role role_list | all | system]
```

各要素の意味は次のとおりです。

- *role_list* − ターゲット・ログインに対して制限する役割のリスト。付与対象者が、このリストのすべての役割を持っていることが必要です。そうでない場合は set proxy コマンドが失敗します。
- *all* − 付与対象者と同じ役割、またはその役割のサブセットを持つユーザについてのみ set proxy を実行できるようにします。
- *system* − 付与対象者がターゲット・ログインと同じシステム役割の組み合わせを持つようにします。

この例は、set proxy をユーザ “joe” に付与しますが、“joe” が ID を、sa、sso、または admin の役割を持つユーザに切り替えることは制限します (ただし、“joe” が既にこれらの役割を持っている場合は、これらの役割を持つユーザに対して set proxy を実行できます)。

```
grant set proxy to joe
restrict role sa_role, sso_role, admin_role
```

“joe” が admin_role を持つユーザ (この例では Our_admin_role) に ID を切り替えようとした場合、joe が admin_role を持っていない限りコマンドは失敗します。

```
set proxy Our_admin_role
Msg 10368, Level 14, State 1:
Server 's', Line 2:自分にない役割がターゲット・ログインに含まれ、その使用を制限されているために、Set session 権限のパーミッションが拒否されました。
```

ユーザ “joe” が admin_role を付与された後でコマンドを再試行すると成功します。

```
grant role admin_role to joe
set proxy Our_admin_role
```

set proxy コマンドの詳細については、『リファレンス・マニュアル：コマンド』を参照してください。

代理権限の実行

set proxy または set session authorization を実行するときは、次の規則に従ってください。

- set proxy と set session authorization は、トランザクション内では実行できません。
- ロックされたログインを使用して、他のユーザの代理となることはできません。たとえば、“joseph” がロックされたログインの場合、次のコマンドは許可されません。

```
set proxy "joseph"
```

- set proxy と set session authorization は、実行するユーザが使用許可を持つすべてのデータベースから実行できます。ただし、指定する *login_name* がデータベース内の有効なユーザであるか、データベースに “guest” が定義されている必要があります。
- 許可されるのは 1 レベルだけです。複数のユーザの代理権限を使用する場合は、それぞれの権限の使用を終了するたびに元の ID に戻る必要があります。
- set proxy または set session authorization をプロシージャ内から実行すると、プロシージャの終了時に自動的に元の ID に戻ります。

自分のログインに set proxy または set session authorization を使用するためのパーミッションが付与されている場合は、これらのコマンドを使用して、別のユーザになり代わることができます。構文は次のとおりです。*login_name* は、master..syslogins 内の有効なログイン名です。

set proxy *login_name*

または

set session authorization *login_name*

ログイン名は引用符で囲んでください。

たとえば、“mary” の代理権限を使用するには、次のコマンドを実行します。

```
set proxy "mary"
```

代理権限を設定したら、サーバでの自分のログイン名と、データベースでの自分のユーザ名を確認します。たとえば、自分のログインが “ralph” であり、set proxy 権限が付与されているものと想定します。このとき、データベース pubs2 において、“sallyn” および “rudolph” としていくつかのコマンドを実行します。“sallyn” には、このデータベースでの有効な名前 (“sally”) がありますが、Ralph と Rudolph にはありません。ただし、pubs2 には “guest” ユーザが定義されています。そこで、次のコマンドを実行できます。

```
set proxy "sallyn"
go
use pubs2
go
select suser_name(), user_name()
```

```
go
------------------------------ --------------------
sallyn                         sally
```

Rudolph に変更するには、まず自身の ID に戻ります。これには、次のコマンドを実行します。

```
set proxy "ralph"
select suser_name(), user_name()
go
------------------------------ --------------------
ralph                          guest
```

Ralph は、このデータベース内では “guest” であることに注意してください。

さらに、次のコマンドを実行します。

```
set proxy "rudolph"
go
select suser_name(), user_name()
go
------------------------------ --------------------
rudolph                        guest
```

Rudolph もデータベース内の有効なユーザではないため、このデータベースでは guest になっています。

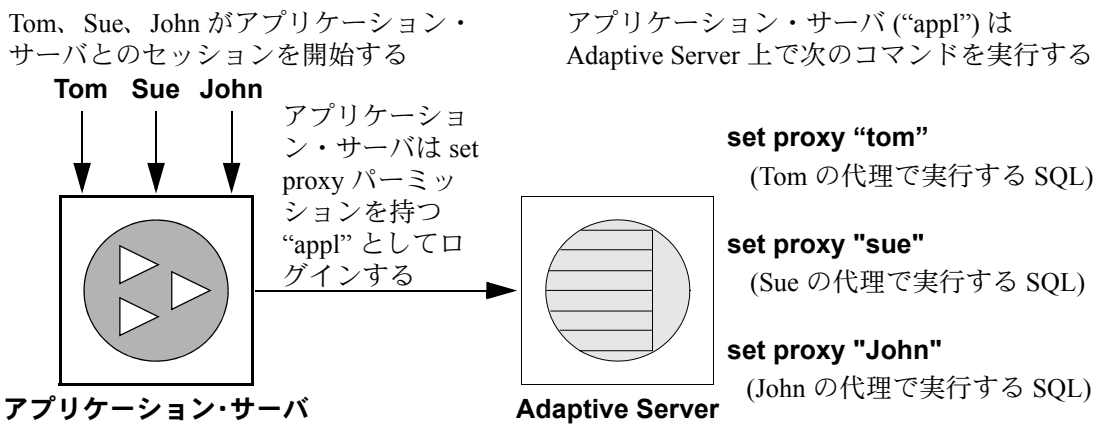
今度は、“sa” アカウントになり代わります。次のコマンドを実行します。

```
set proxy "ralph"
go
set proxy "sa"
go
select suser_name(), user_name()
go
---------------------------- --------------------
sa                           dbo
```

アプリケーションによる代理権限の使用方法

図 6-1 は、汎用ログイン “appl” を使用して Adaptive Server にログインし、多数のユーザに代わってプロシージャとコマンドを実行するアプリケーションを示します。“appl” が Tom になり代わっている間、アプリケーションは Tom のパーミッションを持ちます。同様に、“appl” が Sue と John になり代わると、アプリケーションは、それぞれ Sue と John のパーミッションだけを持ちます。

図 *6-1:* アプリケーションと代理権限



データベース・オブジェクトの所有権の変更

システム・セキュリティ担当者またはデータベース所有者は、alter... modify owner コマンドを使用してデータベース･オブジェクトの所有権を譲渡することができます。

データベース管理者は、このコマンドを使用して従業員の変化に応じてオブジェクトの割り当てを管理したり、データベース・オブジェクトの作成所有権を分割することができます。たとえば、キー管理者が暗号化キーを作成し、その暗号化キーの所有権を別のユーザに譲渡することができます。

サポートしているオブジェクト・タイプ

次に、特定の所有者から所有権を譲渡できるオブジェクトを示します。以下にリストされていないオブジェクトの所有権は変更できません。

所有権を明示的に変更できるオブジェクトは次のとおりです。

- ユーザ・テーブル
- プロキシ・テーブル
- ビュー
- ストアド・プロシージャ
- ユーザ定義関数
- デフォルト

- ルール
- ユーザ定義データ型
- 暗号化キー

所有権を明示的に変更できない従属オブジェクトは次のとおりです。これらのオブジェクトは、明示的に譲渡されるオブジェクトと所有権が同じ場合に暗黙的に譲渡されます。

- トリガ

  トリガが dbo 以外が所有するテーブルまたはビューに対して作成されている場合、そのトリガが所有する dbo の所有権は変更できません。

- テーブルまたはビューの作成時に定義される宣言オブジェクト
  - デフォルト
  - Decrypt_defaults
  - 検査制約
  - 参照制約
  - 分割条件
  - 計算カラム

権限

- システム・セキュリティ担当者には、所有権の譲渡がサポートされるすべてのオブジェクトについて、所有権を譲渡するための権限があります。
- データベースの所有者にはオブジェクト (暗号化キーを除く) の所有権を譲渡する権限がありますが、次の制約があります。
  - データベース・オブジェクトの所有者は、データベース所有者が具体的に所有しているオブジェクトの所有権を譲渡できません。

    オブジェクトにデータベース所有者ユーザ ID の sysobjects.uid と、null またはデータベース所有者のユーザ名の sysobjects.loginame がある場合、そのオブジェクトはデータベース所有者が具体的に所有するオブジェクトとして識別されます。
  - データベース所有者にエイリアスが設定されているユーザは、データベース所有者が作成したオブジェクトやそのユーザが具体的に所有するオブジェクトの所有権を譲渡できません。

    データベース所有者が作成するオブジェクトには、sysobjects.loginame に null 値が含まれます。ユーザが具体的に所有するオブジェクトには、sysobjects.loginame にそのユーザのユーザ名が含まれます。

オブジェクトとそれに対応する所有者を検索するには、sp_helpuser を使用します。

所有権の譲渡

所有権の譲渡は、個別のオブジェクトに固有のものでもあるし、複数のオブジェクトを 1 つのコマンドで譲渡できることもあります。明示的に付与されているオブジェクトのパーミッションを保持するには、preserve permissions を使用します。

構文については、『リファレンス・マニュアル：コマンド』の「alter...modify owner」を参照してください。

次の例では、データベースの所有者が john によって所有されているテーブルを eric に譲渡します。

```
alter table john.table_audit modify owner eric
```

john が所有する全テーブルの所有権を eric に譲渡するには、システム・セキュリティ担当者が次を実行します。

```
alter table john.* modify owner eric
```

john が所有する全オブジェクトの所有権を eric に譲渡するには、システム・セキュリティ担当者が次を実行します。

```
alter all john.* modify owner eric
```

システム・データベースのオブジェクトの所有権の譲渡

Sybase が提供および管理する次のシステム・データベース内のオブジェクトの所有権を変更する場合は、注意が必要です。sybsecurity、sybsystemdb、model、sybsystemprocs、sybsyntax、dbccdb、および tempdb。spt_ プレフィックスの付いたユーザ・テーブル、sp_ プレフィックスの付いたシステム・ストアド・プロシージャなど、Sybase が提供し管理するシステム・オブジェクトの所有権は変更できません。これらのオブジェクトの所有権を変更すると、システムが使用できなくなる可能性があります。

データベース所有者のオブジェクトの所有権の譲渡

非システム・オブジェクトのデータベース所有者は、dbo.object_name を使用して所有権を譲渡できます。* を使って複数のオブジェクトの所有権を譲渡することはできません。

preserve permissions の使用

オブジェクトに対して明示的に付与されているまたは取り消されているパーミッションをすべて保持するには、preserve permissions を指定します。

たとえば、bill はテーブル bill_table の select パーミッションを mark に付与し、次に、mark がテーブル bill_table への select パーミッションを john に付与したとします。その後、preserve permissions を指定してテーブルの所有権が eric に譲渡された場合、mark と john は bill_table のパーミッションを保持します。

次の例では、システム・セキュリティ担当者が既存の明示的に付与されているパーミッションをすべて保持しつつ、ビュー bill.vw_author の所有権を eric に譲渡します。

```
alter view bill.vw_author_in_ca modify owner eric
    preserve permissions
```

preserve permissions が指定されている場合、暗黙的なパーミッションは保持されません。

たとえば、bill が暗号化カラムのあるテーブル bill.encr_table を所有しており restricted decrypt permission 設定オプションが 1 に設定されているとします。システム・セキュリティ担当者が bill.encr_table についての decrypt パーミッションを bill に付与していれば、bill には自身の所有権によって生じたものを変更、削除、挿入、参照、選択、および更新するパーミッションがあります。また、システム・セキュリティ担当者による明示的な付与によって生成したものを復号化するパーミッションも持っています。システム・セキュリティ担当者が preserve permissions を使って bill.encr_table に対する所有権を eric に譲渡すると、bill は decrypt パーミッション以外のテーブルについてのすべてのパーミッションを失います。

preserve permissions が指定されていない場合、所有権を譲渡すると、以前の所有者は、所有権によって暗黙的に生じたオブジェクトについてのパーミッションを失います。新しい所有者は、オブジェクトの所有権を与えられることで暗黙的にパーミッションを得ます。

注意　復号化のパーミッションなど所有権からは派生しないパーミッションの場合、システム・セキュリティ担当者またはデータベース所有者は、オブジェクトのパーミッションを新しい所有者に再度明示的に付与する必要があります。

セキュリティに関する情報

システム・セキュリティ担当者またはデータベース所有者は、可能性として考えられるセキュリティ問題に注意してください。

たとえば、alice は Accounting データベースのユーザで支払給与データへのアクセス権は持っていないとします。彼女は Accounting.dbo.payroll から氏名と給与を選択するプロシージャを作成し、alicep の実行パーミッションを public に付与します。システム・セキュリティ担当者が、preserve permissions オプションで支払給与データへのアクセス権を持つ bill に alicep の所有権を誤って変更すると、所有権の変更後にすべてのパーミッションが保持されるように設定されているため、悪意で alicep プロシージャを実行することによってすべてのユーザが支払給与情報にアクセスできるようになります。

権限なく使用されることを防ぐために、システム・セキュリティ担当者またはデータベース所有者は sp_helprotect を使用してオブジェクトについての既存のパーミッションを確認することができます。

暗号化キーの所有権の譲渡

システム・セキュリティ担当者およびキー所有者は、alter encryption key または alter... modify owner を使用して暗号化キーを譲渡することができます。

alter encryption key コマンドの詳細については、『リファレンス・マニュアル：コマンド』を参照してください。

暗号化キー・コピーの所有者

alter... modify owner コマンドを使用すると、キー・コピーを割り当てられているユーザが暗号化キーの新しい所有者になれなくなります。

暗号化キーの所有権を変更しても、キー・コピーの割り当て対象ユーザは変わりません。たとえば、ユーザ bill が bill.encrkey という名前の暗号化キーを所有し、このキーのキー・コピーを作成して mark に割り当てるとします。bill が bill.encrkey の所有権を eric に譲渡した後も、mark は bill.encrkey のコピーを所有します。

パーミッションを表示する方法

表 6-3 は、代理パーミッション、オブジェクト作成パーミッション、オブジェクト・アクセス・パーミッションに関する情報をレポートするためのシステム・プロシージャを示します。

表 6-3: パーミッションについてレポートするシステム・プロシージャ

レポートする情報	使用
代理	システム・テーブル
ユーザとプロセス	sp_who
データベース・オブジェクトまたはユーザに対するパーミッション	sp_helprotect
特定のテーブルに対するパーミッション	sp_table_privileges
テーブル内の特定のカラムに対するパーミッション	sp_column_privileges

代理権限に対する *sysprotects* テーブルの問い合わせ

ユーザ、グループ、役割に付与されているパーミッションや取り消されたパーミッションに関する情報を表示するには、sysprotects テーブルを問い合わせます。action カラムは、パーミッションを表します。たとえば、set proxy と set session authorization の action の値はどちらも 167 です。

たとえば、次のクエリを実行します。

```
select * from sysprotects where action = 167
```

このクエリを実行すると、パーミッションを付与または取り消したユーザのユーザ ID (grantor カラム)、パーミッションを持つユーザのユーザ ID (uid カラム)、保護のタイプ (protecttype カラム) が表示されます。protecttype カラムには、次の値が含まれます。

- grant with grant を示す 0
- grant を示す 1
- revoke を示す 2

sysprotects テーブルの詳細については、『リファレンス・マニュアル：ビルディング・ブロック』を参照してください。

ユーザとプロセスに関する情報の表示方法

sp_who は、現在のすべての Adaptive Server ユーザとプロセスに関する情報、または特定のユーザやプロセスに関する情報を表示します。sp_who の実行結果の中に、loginame と orignameがあります。ユーザが代理権限のもとで操作を行っている場合、origname には元のログイン名が表示されます。たとえば、“ralph” が次のコマンドを実行してから、いくつかの SQL コマンドを実行するとします。

```
set proxy susie
```

sp_who は、loginame として “susie” を返し、origname として “ralph” を返します。

sp_who は、masater..sysprocesses システム・テーブルを問い合わせます。このテーブルには、サーバ・ユーザ ID のカラム (suid) と元のサーバ・ユーザ ID のカラム (origsuid) があります。

詳細については、『リファレンス・マニュアル：プロシージャ』の「sp_who」を参照してください。

データベース・オブジェクトまたはユーザに対するパーミッション

データベース・オブジェクトまたはユーザごとのパーミッションについて表示するには、sp_helprotect を使用します。指定したオブジェクトのユーザごとのパーミッションを表示することもできます。このプロシージャは、すべてのユーザが実行できます。構文は次のとおりです。

```
sp_helprotect [name [, username [, "grant"
    [,"none"|"granted"|"enabled"|role_name]]]]
```

各パラメータの意味は、次のとおりです。

- *name* は、テーブル、ビュー、またはストアド・プロシージャの名前、あるいは現在のデータベース内のユーザ、グループ、または役割の名前です。名前を指定しないで sp_helprotect を実行すると、データベース内のすべてのパーミッションが表示されます。
- *username* は、現在のデータベース内のユーザの名前です。

  username を指定すると、指定したオブジェクトに対してそのユーザに付与されているパーミッションだけがレポートされます。*name* がオブジェクトではない場合は、sp_helprotect を実行すると *name* がユーザ、グループ、役割のどれに該当するかが検査され、これらのいずれかである場合は、そのユーザ、グループ、または役割に対するパーミッションが表示されます。キーワード grant を指定し、*name* にオブジェクト以外のものを指定して sp_helprotect を実行すると、with grant option によって付与されたすべてのパーミッションが表示されます。
- grant を指定すると、with grant option で *name* に付与されているパーミッションが表示されます。

- none を指定すると、ユーザに付与されている役割は無視されます。
- granted を指定すると、ユーザに付与されているすべての役割に関する情報も表示されます。
- enabled を指定すると、そのユーザがアクティブ化したすべての役割に関する情報も表示されます。
- *role_name* を指定すると、この役割がユーザに付与されているかどうかには関係なく、指定した役割に関するパーミッション情報だけが表示されます。

たとえば、次の一連の grant 文と revoke 文を発行するとします。

```
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
```

Judy が titles テーブルの各カラムに対して現在持っているパーミッションを調べるには、次のように入力します。

```
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
```

最初の行は、データベース所有者 (“dbo”) が Judy に titles テーブルのすべてのカラムを選択できるパーミッションを付与していることを示します。残りの行は、Judy は表示されているカラムの更新だけができることを示しています。つまり、他のユーザに select パーミッションや update パーミッションを付与することはできません。

publishers テーブルに対する Judy のパーミッションを調べるには、次のように入力します。

```
sp_helprotect publishers, judy
```

次の表示では、grantable カラムの値が TRUE です。つまり、Judy は他のユーザにパーミッションを付与できます。

```
grantor grantee type   action   object     column      grantable
------- ------ -----   ------   ------     ------      -------
dbo     judy    Grant   Select  publishers   all         TRUE
```

特定のテーブルに対するパーミッションを表示する方法

指定したテーブルに関するパーミッション情報を表示するには、sp_table_privileges を使用します。構文は次のとおりです。

```
sp_table_privileges table_name [, table_owner
    [, table_qualifier]]
```

各パラメータの意味は、次のとおりです。

- *table_name* － テーブルの名前です。これは必須です。
- *table_owner* は、テーブル所有者が “dbo” でも sp_table_privileges を実行するユーザでもない場合に、テーブル所有者の名前を指定するのに使用します。
- *table_qualifier* は現在のデータベースの名前です。

省略するパラメータには、null を使用します。

たとえば、次の文は、titles テーブルについて付与されているすべてのパーミッションの情報を返します。

```
sp_table_privileges titles
```

sp_table_privileges の出力の詳細については、『リファレンス・マニュアル：プロシージャ』を参照してください。

特定のカラムに対するパーミッションを表示する方法

テーブル内のカラムに対するパーミッションに関する情報を表示するには、sp_column_privileges を使用します。構文は次のとおりです。

```
sp_column_privileges table_name [, table_owner
    [, table_qualifier [, column_name]]]
```

各パラメータの意味は、次のとおりです。

- *table_name* はテーブルの名前です。
- *table_owner* は、テーブル所有者が “dbo” でも sp_column_privileges を実行するユーザでもない場合に、テーブル所有者の名前を指定するのに使用します。
- *table_qualifier* は現在のデータベースの名前です。
- *column_name* には、パーミッション情報を表示するカラムの名前を指定します。

省略するパラメータには、null を使用します。

たとえば、次の文は、publishers テーブルの pub_id カラムについての情報を返します。

```
sp_column_privileges publishers, null, null, pub_id
```

sp_column_privileges の出力の詳細については、『リファレンス・マニュアル：プロシージャ』を参照してください。

セキュリティ・メカニズムとしてのビューとストアド・プロシージャの使用

ビューとストアド・プロシージャは、セキュリティ・メカニズムとして使用できます。ビューやストアド・プロシージャを使用することにより、ユーザがデータに直接アクセスできないようにして、データベース・オブジェクトへのユーザのアクセスを制御することができます。たとえば、projects テーブル内のコスト情報を更新するプロシージャに対する execute パーミッションを担当者に付与すれば、そのテーブル内の機密データをユーザが参照できないようにすることができます。この機能を活用するには、ビューやストアド・プロシージャを作成するユーザが、そのプロシージャやビューだけでなく、基本となるオブジェクトも所有する必要があります。基本となるオブジェクトを所有していない場合は、ビューやストアド・プロシージャを利用するユーザが、そのオブジェクトにアクセスするためのパーミッションを持っていなければなりません。パーミッションが必要な場合の詳細については、「所有権の連鎖の理解」(204 ページ) を参照してください。

ビューまたはプロシージャを使用するとき、必要に応じて Adaptive Server によるパーミッションの検査が行われます。ビューまたはプロシージャを作成するときには、基本となるオブジェクトに対するパーミッション検査は行われません。

セキュリティ・メカニズムとしてのビューの使用

ビューを使用すれば、ユーザは自分が参照可能なデータだけを問い合わせたり変更したりできます。ビューに定義されていないデータベースの部分は、参照することも、アクセスすることもできません。

ビューにアクセスするためのパーミッションの付与や取り消しは、ビューの基本となるテーブルに対するパーミッションとは無関係に、明示的に行う必要があります。ビューと基本となるテーブルの所有者が同じである場合は、基本となるテーブルに対するパーミッションを付与する必要はありません。ビューへのアクセスが許可されていても、その基本となるテーブルへのアクセスが許可されていないユーザは、基本となるテーブルのうち、ビューに含まれていないデータを参照することはできません。

複数のビューを定義して、そのビューに対してパーミッションを選択的に付与すれば、ユーザまたはユーザの組み合わせごとにアクセス可能なデータのサブセットを設定することができます。アクセスは次のように制限できます。

- アクセスをベース・テーブルのローのサブセット (値に依存するサブセット) に制限できます。たとえば、ビジネスと心理学の本のローだけを含むビューを定義して、その他のタイプの本についての情報を一部のユーザから見えないようにすることができます。

- アクセスを、ベース・テーブルのカラムのサブセット (値に依存しないサブセット) に制限できます。たとえば、titles テーブルのすべてのローが含まれるが、機密情報に属する印税 (price) と前払い額 (advance) のカラムを除いたビューを定義できます。
- アクセスを、ベース・テーブルのローとカラムのサブセットに制限できます。
- アクセスを、複数のベース・テーブルのジョインの条件を満たすローに制限できます。たとえば、titles テーブル、authors テーブル、titleauthor テーブルをジョインするビューを定義します。このビューは、作家についての個人的な情報や、その本についての金銭的な情報は表示しません。
- アクセスをベース・テーブル内のデータの統計情報に制限できます。たとえば、本のタイプごとの平均価格だけが表示されるビューを定義します。
- アクセスを別のビューのサブセット、またはビューとベース・テーブルの組み合わせのサブセットに制限できます。

たとえば、一部のユーザを、金銭と売上に関する titles テーブル内のカラムにアクセスできないようにしたいと仮定します。その場合には、金銭と売上に関するカラムを除いて titles テーブルのビューを作成し、そのビューに対するパーミッションをすべてのユーザに付与して、テーブルに対するパーミッションは営業部門にだけ付与します。

```
grant all on bookview to public
grant all on titles to sales
```

これらの権限の条件をビューを使わずに設定するには、次の文を使用します。

```
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
```

この 2 番目の方法を使用した場合は、sales グループのメンバでないユーザが select * from titles コマンドを入力したときに、次の語句が含まれるメッセージが突然表示されて混乱を招くおそれがあります。

```
パーミッションが拒否されました
```

このアスタリスクは、titles テーブル内のすべてのカラムのリストに展開されます。このリスト内のカラムのいくつかについては、営業部門以外のユーザからはパーミッションが取り消されているので、そのカラムに対するアクセスは拒否されます。ユーザがアクセス権を持っていないカラムがエラー・メッセージに表示されます。

営業部門以外のユーザが、パーミッションを持つすべてのカラムを表示するには、カラムを明示的に指定する必要があります。このため、ビューを作成して、適切なパーミッションを付与する方が簡単です。

ビューを使用すると、「コンテキストで区別されるプロテクション」を実現することもできます。たとえば、データ入力者に、自分が追加または更新したローだけにアクセスできるパーミッションを付与するビューを作成します。これを行うには、テーブルにカラムを追加し、各ローを入力したユーザのユーザ ID が自動的にデフォルト値によってそのカラムに記録されるようにします。このデフォルト値は、create table 文で次のように定義します。

```
create table testtable
    (empid       int,
     startdate   datetime,
     username     varchar(30) default user)
```

次に、このテーブルのローのうち、uid が現在のユーザに等しいローがすべて表示されるビューを定義します。

```
create view context_view
as
    select *
    from testtable
    where username = user_name()
with check option
```

このビューによって検索できるローは、ビューに対して select コマンドを発行するユーザの ID によって異なります。ビュー定義に with check option を追加すると、データ入力者が username カラム内の情報を改ざんできないようにすることができます。

セキュリティ・メカニズムとしてのストアド・プロシージャの使用

ストアド・プロシージャの所有者と、基本となるすべてのオブジェクトの所有者が同じならば、プロシージャを使うためのパーミッションを所有者が他のユーザに付与するときに、基本となるオブジェクトに対するパーミッションを付与する必要はありません。たとえば、指定されたテーブルのローとカラムのサブセットを更新するストアド・プロシージャを実行するためのパーミッションをユーザに付与するとき、ユーザはそのテーブルに対するその他のパーミッションを持っていなくてもかまいません。

役割とストアド・プロシージャ

grant execute コマンドを使うと、ストアド・プロシージャに対する実行パーミッションを、指定した役割を付与されているすべてのユーザに付与できます。同様に revoke execute コマンドを使って、このパーミッションを削除できます。ただし、grant execute パーミッションによる方法では、特定の役割を持たないユーザにストアド・プロシージャの実行パーミッションが付与されることを防ぐことはできません。

セキュリティをさらに高めるには、プロシージャ内で has_role システム関数を使うことによって、役割を付与されているユーザだけがそのプロシージャを実行できるように制限できます。ユーザに特定の役割 (sa_role、sso_role、oper_role、または任意のユーザ定義の役割) が付与されている場合は has_role は 1 を返し、付与されていない場合は 0 を返します。たとえば、has_role を使用して、ユーザがシステム管理者の役割を持っているかどうかを確認するプロシージャを次に示します。

```
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)
```

has_role の詳細については、『リファレンス・マニュアル：ビルディング・ブロック』の「システム関数」を参照してください。

所有権の連鎖の理解

ビューは別のビューやテーブルに従属します。プロシージャは別のプロシージャ、ビュー、またはテーブルに従属します。このような従属性を「所有権の連鎖」と考えることができます。

通常は、ビューの所有者はその基本となるオブジェクト (他のビューやテーブル) も所有します。ストアド・プロシージャの所有者は、そのプロシージャによって参照されるすべてのプロシージャ、テーブル、ビューを所有します。

ビューとその基本となるオブジェクトは、ストアド・プロシージャとそれが参照するすべてのオブジェクトと同様に、通常はすべて同じデータベース内に存在しますが、これは必須ではありません。これらのオブジェクトが別のデータベース内に存在する場合は、ビューまたはストアド・プロシージャを使用するユーザは、オブジェクトが存在するすべてのデータベース内の有効なユーザか guest ユーザである必要があります。このため、データベース所有者による許可を受けなければ、ユーザはデータベースにアクセスできません。

プロシージャまたはビューに対する execute パーミッションを付与されているユーザがそのプロシージャまたはビューを使用するときに、次の条件に該当する場合は、基本となるオブジェクトのパーミッションの検査は一切行われません。

- これらのオブジェクトとビューまたはプロシージャが同じユーザによって所有されている場合。

- ビューまたはプロシージャにアクセスするユーザが、基本となるオブジェクトが存在するそれぞれのデータベース内の有効なユーザか guest ユーザである場合。

ただし、すべてのオブジェクトの所有者が同じでない場合は、所有権の連鎖が切れたところでオブジェクトのパーミッションの検査が行われます。つまり、オブジェクト A がオブジェクト B を参照していて、オブジェクト A の所有者とオブジェクト B の所有者が異なる場合は、オブジェクト B に対するパーミッションが検査されます。このようにして、データへのアクセスをどのユーザに許可するかという制御を元のデータの所有者が維持できるようにします。

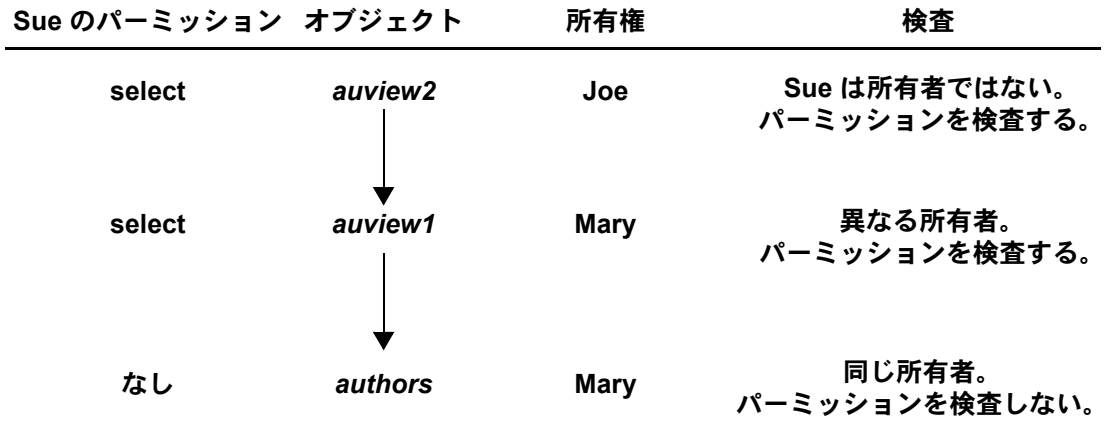
通常は、ビューを作成するユーザが注意しなければならないのは、そのビューに対するパーミッションの付与だけです。たとえば、Mary が、自分が所有する authors テーブルに auview1 というビューを作成したとします。Mary が auview1 に対する select パーミッションを Sue に付与すると、Sue がこのビューにアクセスするとき、authors に対するパーミッションの検査は行われません。

ただし、別のユーザが所有しているオブジェクトに従属するビューまたはストアド・プロシージャを作成する場合は、自分が付与するパーミッションが、それらの他の所有者によって許可されているパーミッションに従属することに注意してください。

ビューと所有権の連鎖の例

Joe が作成する auview2 というビューが、Mary のビュー auview1 に従属するとします。Joe は auview2 に対する select パーミッションを Sue に付与します。

図 6-2: ビューの所有権の連鎖とパーミッション検査 (ケース 1)

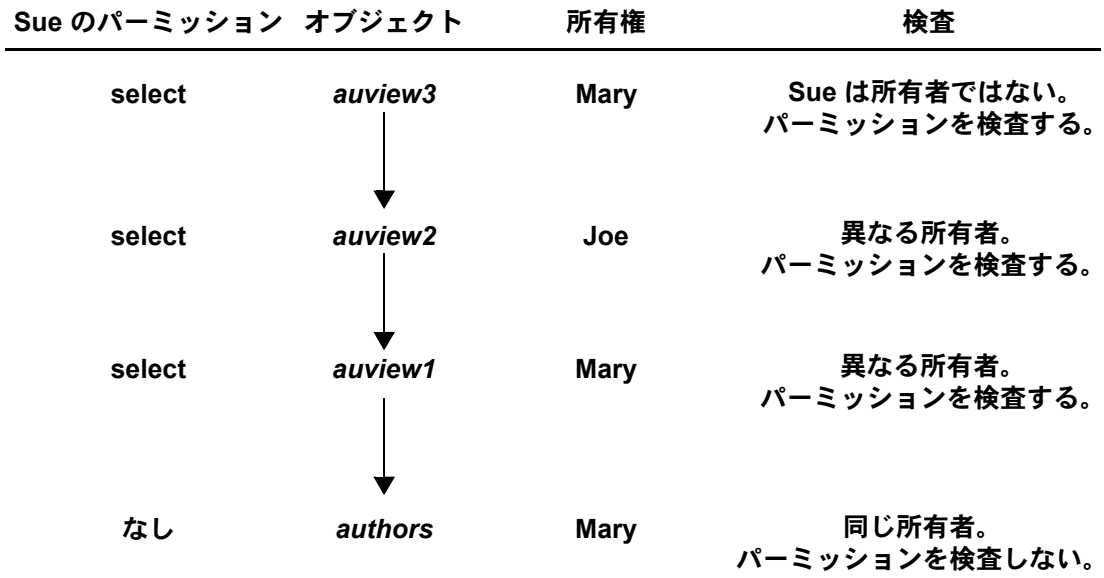


Adaptive Server は auview2 と auview1 に対するパーミッションを検査して、Sue がこれらのビューを使用できると判断します。また、auview1 と authors に対する所有権を検査して、これらの所有者が同じであると判断します。したがって、Sue は auview2 を使用できます。

この例をさらに一歩進めて、Joe のビュー auview2 が auview1 に従属していて、auview1 が authors に従属しているとします。Mary は、Joe の auview2 の上に auview3 を作成します。auview1 と authors は Mary によって所有されます。

所有権の連鎖は次のようになります。

図 6-3: ビューの所有権の連鎖とパーミッション検査 (ケース 2)

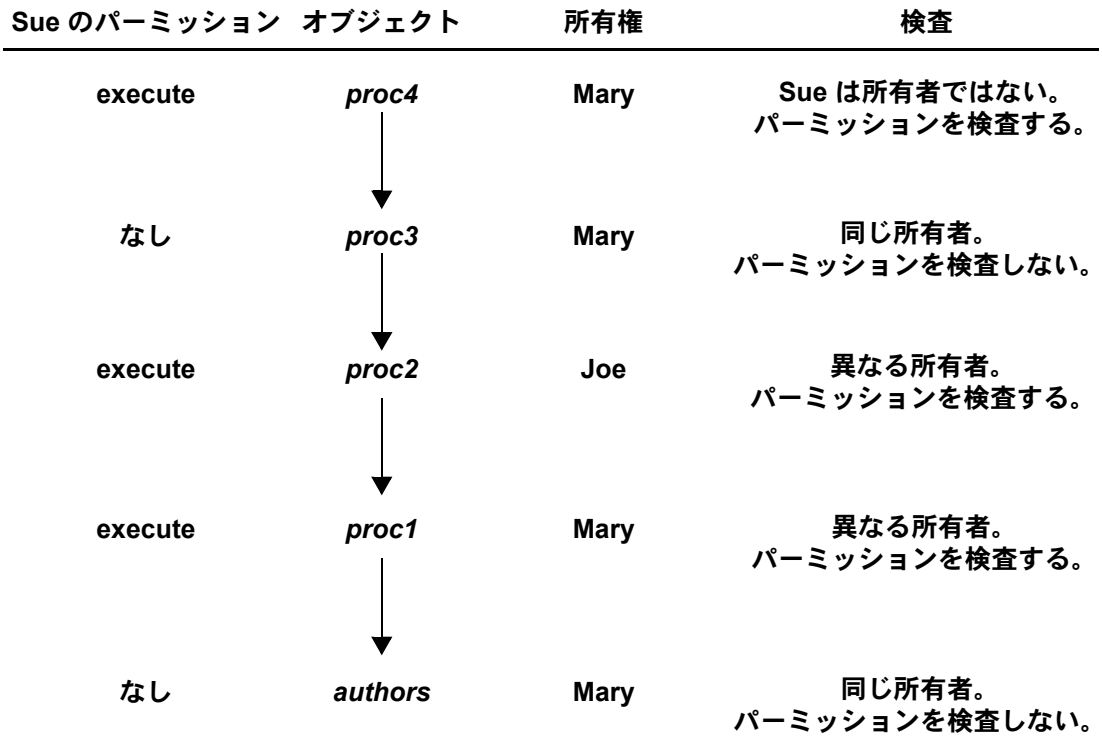


Sue が auview3 にアクセスすると、Adaptive Server は、auview3、auview2、auview1 に対するパーミッションを検査します。auview2 に対するパーミッションが Joe から Sue に付与され、auview3 と auview1 に対するパーミッションが Mary から付与されていれば、Adaptive Server はアクセスを許可します。Adaptive Server によってパーミッションの検査が行われるのは、連鎖内の直前のオブジェクトが別の所有者によって所有されている場合 (またはそのオブジェクトが連鎖内の最初のオブジェクトである場合) だけです。たとえば、auview2 は検査の対象です。これは、直前のオブジェクト auview3 が別のユーザによって所有されているからです。authors に対するパーミッションは検査されません。authors に直接従属しているオブジェクト auview1 が同じユーザによって所有されているからです。

プロシージャと所有権の連鎖の例

プロシージャはビューと同じ規則に従います。たとえば、所有権の連鎖が次のようになっているとします。

図 6-4: ストアド・プロシージャの所有権の連鎖とパーミッション検査



Sue が proc4 を実行するには、proc4、proc2、proc1 を実行するためのパーミッションが必要です。proc3 は proc4 と所有者が同じなので、proc3 を実行するためのパーミッションは必要ありません。

Adaptive Server は、Sue が proc4 を実行するたびに、proc4 とこのプロシージャが参照するすべてのオブジェクトに対する Sue のパーミッションを検査します。Adaptive Server は、参照されるオブジェクトのうちどれを検査するかを把握しています。この情報は Sue が proc4 を初めて実行したときに決定され、プロシージャの実行プランとともに保管されています。プロシージャによって参照されるオブジェクトが削除されたり再定義されたりしない限り、検査するオブジェクトについての最初の決定は変更されません。

この保護階層を使用すれば、オブジェクトの所有者がそのオブジェクトに対するアクセスを完全に制御できます。所有者は、テーブルへのアクセスだけではなく、ビューやストアド・プロシージャへのアクセスも制御できます。

トリガのパーミッション

「トリガ」は、整合性、特に参照整合性を保つために使用される特別な種類のストアド・プロシージャです。トリガは直接実行されることはなく、テーブルの変更の結果として実行されます。トリガに対するパーミッションを付与 (grant) または取り消す (revoke) 方法はありません。

オブジェクトに対してトリガを作成できるのは、そのオブジェクトの所有者だけです。ただし、テーブルに対するトリガが、別のユーザによって所有されているオブジェクトを参照する場合は、所有権の連鎖が切れることになります。プロシージャに適用される保護階層規則はトリガに対しても適用されます。

トリガが影響を与えるオブジェクトは、通常はそのトリガを所有するユーザが所有するオブジェクトですが、別のユーザが所有するオブジェクトを変更するトリガを作成することもできます。この場合は、トリガをアクティブにする方法でオブジェクトを変更するユーザはすべて、他のオブジェクトに対するパーミッションも持っている必要があります。

トリガが影響を与えるオブジェクトに対するパーミッションがユーザに付与されていないという理由で Adaptive Server がデータ変更コマンドに対するパーミッションを拒否した場合は、データ変更トランザクション全体がロールバックされます。

詳細については、『Transact-SQL ユーザーズ・ガイド』の「第 19 章 トリガ：参照整合性」を参照してください。

ロー・レベル・アクセス制御の使用

ロー・レベル・アクセス制御には次の機能があり、データベース所有者やテーブル所有者は安全なデータ・アクセス環境を自動的に作成できます。

- より細密なデータ・セキュリティ。テーブルとカラムだけではなく、個々のローに対してパーミッションを設定できます。
- グループ、役割、アプリケーションに応じた自動データ・フィルタリング。
- サーバでのコード化によるデータ・レベルのセキュリティ。

ロー・レベル・アクセス制御の次の 3 つの機能によって、テーブルの個々のローのデータへのアクセスを制御します。

- データベース所有者が定義してテーブルにバインドするアクセス・ルール。
- ユーザ定義のコンテキストを定義、保存、検索するための組み込み関数の集合である Application Context Facility。
- データベース所有者、sa_role、またはユーザが作成できるログイン・トリガ。

Adaptive Server のロー・レベル・アクセス制御はすべてのデータ操作言語 (DML) に適用されるので、ユーザがアクセス制御を回避してデータを取得することはできません。

ロー・レベル・アクセス制御を使用するようにシステムを設定する構文は次のとおりです。

```
sp_configure "enable row level access", 1
```

このオプションを使用するときは、Adaptive Server が使用するメモリの量がわずかに増えます。また、ASE_RLAC ライセンス・オプションが必要です。ロー・レベル・アクセス制御は動的オプションなので、Adaptive Server を再起動する必要はありません。

アクセス・ルール

ロー・レベル・アクセス制御機能を使用するには、既存の create rule の構文に access オプションを追加します。アクセス・ルールは、参照または変更できるローを制限するものです。

アクセス・ルールは、特定のカラムでユーザが挿入または更新できる値をテーブル所有者が制御するためのドメイン・ルールに似ています。ドメイン・ルールは、追加されるデータについて制限を適用するもので、update コマンドと insert コマンドに対して機能します。

アクセス・ルールは、検索されるデータを制限するもので、select、update、delete の各オペレーションに適用されます。アクセス・ルールは、クエリで読み込まれるすべてのカラムに対して適用されます。select リストで指定されていないカラムについても同様です。つまり、特定のクエリにおいて、更新されるテーブルにはドメイン・ルールが適用され、読み込まれるすべてのテーブルにアクセス・ルールが適用されます。

次に例を示します。

```
insert into orders_table
select * from old_orders_table
```

このクエリでは、orders_table に対するドメイン・ルールと old_orders_table に対するアクセス・ルールがある場合に、orders_table は更新されるのでドメイン・ルールが適用され、old_orders_table は読み取られるのでアクセス・ルールが適用されます。

アクセス・ルールを使用することは、ビューを使用することや、where 句のあるアドホック・クエリを使用することに似ています。アクセス・ルールが付加された後でクエリのコンパイルと最適化が行われるので、パフォーマンスが低下することはありません。アクセス・ルールは、テーブル・データの仮想ビューを実現するものです。つまり、カラムにバインドされた特定のアクセス・ルールに応じて変化するビューです。

アクセス・ルールは、sp_addtype を使用して定義するユーザ定義データ型にバインドできます。アクセス・ルールはユーザ・テーブルに対して適用されます。これを利用すれば、テーブル所有者やデータベース所有者が正規化スキーマの中でカラムにアクセス・ルールをバインドするという管理作業を行う必要はありません。たとえば、ベース型が varchar(30) であるユーザ定義データ型を作成して username という名前を付け、このデータ型にアクセス・ルールをバインドしたとします。このアクセス・ルールは、アプリケーション内の username 型のカラムを持つすべてのテーブルに適用されます。

アプリケーション開発者は、Java とアプリケーション・コンテキストを使って柔軟なアクセス・ルールを作成できます。詳細については、「ユーザ定義 Java 関数としてのアクセス・ルール」(214 ページ) と「Application Context Facility の使用」(217 ページ) を参照してください。

アクセス・ルールの構文

アクセス・ルールを作成するには、create rule 構文の access パラメータを使用します。

```
create [or|and] access rule (access_rule_name)
as (condition)
```

アクセス・ルールを持つサンプル・テーブルの作成

この項では、テーブルを作成してアクセス・ルールをバインドするプロセスを示します。

テーブルの作成

テーブル所有者は、テーブル T を作成して (username char(30)、title char(30)、classified_data char(1024))、次のデータを入力します。

```
AA, "Administrative Assistant","Memo to President"
AA, "Administrative Assistant","Tracking Stock Movements"
VP1, "Vice President", "Meeting Schedule"
VP2, "Vice President", "Meeting Schedule"
```

アクセス・ルールの作成とバインド

テーブル所有者は、アクセス・ルール uname_acc_rule を作成して、テーブル T の username カラムにバインドします。

```
create access rule uname_acc_rule
as @username = suser_name()
-----------
sp_bindrule uname_acc_rule, "T.username"
```

テーブルに対するクエリ

次のクエリを発行します。

```
select * from T
```

Adaptive Server は、T のusername カラムにバインドされているアクセス・ルールを処理して、クエリ・ツリーに付加します。次に、このツリーが最適化され、実行プランが生成されて実行されます。このクエリは、アクセス・ルールで指定されているフィルタ句をユーザが指定してクエリを実行したかのように実行されます。つまり、アクセス・ルールが付加されると、次のクエリが実行されることになります。

```
select * from T where T.username = suser_name().
```

条件 `where T.username = suser_name()` の部分は、サーバによって強制的に追加されます。ユーザがこのアクセス・ルールを回避することはできません。

Administrative Assistant がこの select クエリを実行したときの結果は次のとおりです。

```
AA, "Administrative Assistant","Memo to President"
AA, "Administrative Assistant","Tracking Stock Movements"
```

アクセス・ルールの削除

アクセス・ルールを削除する前に、次の例に示すように sp_unbindrule を使用してカラムまたはデータ型へのそのアクセス・ルールのバインドを解除してください。

```
sp_unbindrule "T.username",
NULL, "all"
```

デフォルトでは、sp_unbindrule を実行すると、カラムに付加されているドメイン・ルールのバインドが解除されます。

バインドを解除した後で、アクセス・ルールを削除します。

```
drop rule "rule_name"
```

たとえば、次のように結果が表示されます。

```
drop rule "T.username"
```

拡張アクセス・ルールの構文

アクセス・ルールはそれぞれ 1 つのカラムにバインドされますが、1 つのテーブルで複数のアクセス・ルールを使用できます。create rule には、複数のアクセス・ルールの評価を処理するための AND パラメータと OR パラメータがあります。AND アクセス・ルールと OR アクセス・ルールを作成するには、拡張アクセス・ルールの構文を使用します。

- AND アクセス・ルール

  create and access rule rule_name

- OR アクセス・ルール

  create or access rule rule_name as

AND アクセス・ルールと OR アクセス・ルールは、カラムまたはユーザ定義のデータ型にバインドできます。拡張アクセス・ルールの構文を使用すると、複数のアクセス・ルールを同じテーブルにバインドできますが、カラムごとにバインドできるアクセス・ルールは 1 つだけです。ユーザがテーブルにアクセスすると、アクセス・ルールが有効になり、デフォルトでは AND アクセス・ルールが先にバインドされ、次に OR アクセス・ルールがバインドされます。

複数のアクセス・ルールをテーブルにバインドするときに、AND と OR のどちらも指定していない場合のデフォルトのアクセス・ルールは AND となります。

テーブルのローに対するアクセス・ルールが 1 つだけで、そのルールが OR アクセス・ルールとして定義されている場合は、AND アクセス・ルールとして動作します。

アクセス・ルールと拡張アクセス・ルールの使用

アクセス・ルールの作成

次の手順で、アクセス・ルールを作成します。

```
create access rule empid1_access
as @empid = 1

create access rule deptno1_access
as @deptid = 2
```

次の手順で、OR アクセス・ルールを作成します。

```
create or access rule name1_access
as @name = "smith"

create or access rule phone_access
as @phone = "9999"
```

テーブルの作成

次の手順で、テスト・テーブルを作成します。

```
create table testtab1 (empno int, deptno int,name char(10),
phone char(4))
```

テーブルへのルールのバインド

次の手順で、テスト・テーブルのカラムにアクセス・ルールをバインドします。

```
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)
```

テーブルへのデータの挿入

次の手順で、テスト・テーブルに値を挿入します。

```
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)
```

アクセス・ルールの例

次の例では、アクセス・ルールによって返されるローの内容が、どのようにアクセス・ルールによって制限されているかを示します。

例 1　2 つのローからの情報を返します。

```
/* 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)
```

例 2　4 つのローからの情報を返します。

```
/* 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
```

```
(4rowsaffected)

/* unbind all deptno rules from specific column */
sp_unbindrule "testtab1.deptno",NULL,"all"
/*Rule unbound from table column.*/

(return status = 0)
```

例 3　6 つのローからの情報を返します。

```
/* 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
```

アクセス・ルールと alter table コマンド

テーブル所有者が alter table コマンドを実行するとき、コマンド実行中はアクセス・ルールは無効になり、コマンド実行終了時に再び有効化されます。アクセス・ルールが無効化されるのは、alter table コマンドの実行中にテーブル・データをフィルタしないようにするためです。

アクセス・ルールと *bcp*

コマンド bcp を使ってテーブルからデータをコピーするときは、アクセス・ルールが適用されます。alter table の場合とは異なり、Adaptive Server がアクセス・ルールを無効にすることはできません。これは、bcp はテーブルに対する選択パーミッションを持つユーザであれば誰でも使用できるためです。

セキュリティのために、データベース所有者は、バルク・コピー・アウトの実行中はテーブルを排他的にロックし、アクセス・ルールを無効にします。アクセス・ルールが無効化されている間は、ロックによって他のユーザのアクセスを不可能にします。データベース所有者は、データのコピーが完了したら、アクセス・ルールをバインドし、テーブルのロックを解除します。

ユーザ定義 Java 関数としてのアクセス・ルール

アクセス・ルールでは、ユーザ定義 Java 関数を使用できます。たとえば、アプリケーションのプロファイル、アプリケーションにログインしたユーザ、アプリケーションを実行するために現在ユーザに与えられている役割などを使用する高度なルールを作成する場合に、Java 関数を使用します。

次の GetSecVal メソッドを使用する Java クラスでは、JDBC を使用する Java メソッドをユーザ定義関数としてアクセス・ルール内で使用する方法を示します。

```
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);
}
}
}
```

次のように、この Java コードのコンパイル後、同じプログラムを isql から実行できます。

たとえば、次のように結果が表示されます。

```
javac sec_class.java
jar cufo sec_class. jar sec_class.class
installjava -Usa -Password -f/work/work/FGAC/sec_class.jar -
-D testdb
```

isql で、次のように入力します。

```
/*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
```

Application Context Facility の使用

データベース・サーバ上のアプリケーションは、データへのアクセスを制限する必要があります。アプリケーションのコーディングにあたっては、ユーザのプロファイルを十分考慮します。たとえば、人事アプリケーションは、どのユーザに給与データの更新が許可されているかを認識するように作成します。

このようなコーディングを可能にする属性によって、アプリケーション・コンテキストが構成されます。Application Context Facility (ACF) は 3 つの組み込み関数で構成されており、セッション内でユーザに割り当てられた固有値との比較をアクセス・ルールの中で実行できるようにすることによって、安全なデータ・アクセス環境を実現します。

アプリケーション・コンテキストは、context_name、attribute_name、attribute_value から構成されます。ユーザは、各コンテキストに対してコンテキスト名、属性、値を定義します。Sybase が提供するデフォルトの読み込み専用アプリケーション・コンテキスト SYS_SESSION を使用すると、セッション固有の情報にアクセスできます。このアプリケーション・コンテキストの説明は、表 6-4 (224 ページ) を参照してください。また、「アプリケーション・コンテキストの作成と使用」(220 ページ) で説明しているように、ユーザが独自のアプリケーション・コンテキストを作成することもできます。

ユーザ・プロファイルとアプリケーション・プロファイル (システム管理者が作成するテーブルで定義される) を組み合わせることにより、複数のセキュリティ方式の累積や重ね合わせが可能となります。

ACF を使用すると、ユーザは次のものを定義、保存、検索できます。

- ユーザ・プロファイル (ユーザに付与された役割、およびユーザが属するグループ)
- 現在使用されているアプリケーション・プロファイル

1 つのセッションで使用できるアプリケーション・コンテキストの数に制限はありません。また、1 つのコンテキストで定義できる属性と値のペアの数も制限はありません。ACF コンテキストのローは 1 つのセッションに固有であり、複数のセッションにわたっては存続しません。ただし、ローカル変数とは異なり、ネストした文が実行されるときも、レベルを越えて利用可能です。ACF は、このようなコンテキスト・ローを設定、取得、検索、削除する組み込み関数の集まりです。

アプリケーション・コンテキスト関数を使ってパーミッションを設定する

アプリケーション・コンテキスト関数は、select 文の中で実行します。関数の所有者はサーバのシステム管理者です。アプリケーション・コンテキストを作成、設定、検索、削除するには、組み込み関数を使用します。

この関数で使用されるデータは、全テーブルに対する全ログインのデータを含むテーブルで定義されます。このテーブルは、システム管理者によって作成されます。このテーブルの詳細については、「ログイン・トリガの使用」(226 ページ) を参照してください。

- set_appcontext() は保存を実行します。

  ```
  select set_appcontext ("titles", "rlac", "1")
  ```

- get_appcontext() に、セッション内のコンテキストの 2 つの要素を渡すと、3 つ目の要素が返されます。

```
select get_appcontext ("titles", "rlac")
------------------------
1
```

これらの関数および list_appcontext と rm_appcontext の詳細については、「アプリケーション・コンテキストの作成と使用」(220 ページ) を参照してください。

権限の付与と取り消し

特定のデータベース内のオブジェクトに対するアクセス権限を、ユーザ、役割、グループに付与したり取り消したりすることができます。ただし、create database、set session authorization および connect のみは例外です。これらの権限を付与されるユーザは master データベースの有効なユーザでなければなりません。他の権限を使用するには、そのオブジェクトが存在するデータベースの有効なユーザでなければなりません。

関数を使用するということは、特別な処置をとらない限り、ログインしたユーザがそのセッションのプロファイルを再設定できてしまうということです。Adaptive Server は組み込み関数を監査しますが、問題に気づく前にセキュリティが損なわれている可能性もあります。これらの組み込み関数へのアクセスを制限するには、権限の grant と revoke を使用します。sa_role を付与されたユーザだけが、組み込み関数に対する権限の付与と取り消しを実行できます。関数によって実行される、サーバによる強制データ・アクセス・コントロール・チェックの中では、select 権限のみがチェックされます。

有効なユーザ

関数にはオブジェクト ID はなく、ホーム・データベースもありません。したがって、各データベースの所有者は、関数に対する select 権限を該当するユーザに付与する必要があります。Adaptive Server は、ユーザのデフォルト・データベースを特定して、そのデータベースに対するパーミッションをチェックします。この方法では、データベース所有者による select 権限の付与が必要となるのはユーザのデフォルト・データベースだけです。他のデータベースについても制限が必要な場合は、そのデータベースの所有者が明示的にそのデータベースでのユーザの権限を取り消す必要があります。

関数に対する権限の付与や取り消しを行うときに、ユーザのデータ・アクセス制御チェックが行われるのは、アプリケーション・コンテキスト組み込み関数だけです。他の関数への権限の付与や取り消しを行っても Adaptive Server には何の影響も与えません。

public に付与されている権限の影響を受けるのは、システム管理者が作成するテーブルで指定されたユーザだけです。このテーブルの詳細については、「ログイン・トリガの使用」(226 ページ) を参照してください。guest ユーザが権限を持つのは、sa_role がこのテーブルに追加することによって明示的に権限を与えた場合だけです。

システム管理者は、以下のコマンドを実行して、特定のアプリケーション・コンテキスト関数に対する select 権限を付与または取り消します。

- grant select on set_appcontext to user_role
- grant select on set_appcontext to joe_user
- revoke select on set_appcontext from joe_user

アプリケーション・コンテキストの作成と使用

アプリケーション・コンテキストの作成と管理に利用できる関数は以下のとおりです。詳細については、『リファレンス・マニュアル：ビルディング・ブロック』を参照してください。

- set_appcontext
- get_appcontext
- list_appcontext
- rm_appcontext

set_appcontext

指定されたユーザ・セッションのアプリケーション・コンテキスト名、属性名、属性値を設定します。これらは、アプリケーションの属性によって定義されます。

```
set_appcontext ("context_name", "attribute_name", "attribute_value")
```

パラメータ

- *context_name* − アプリケーション・コンテキスト名を指定するロー。データ型 char(30) として保存されます。
- *attribute_name* − アプリケーション・コンテキスト属性名を指定するロー。データ型 char(30) として保存されます。
- *attribute_value* − アプリケーション属性値を指定するロー。データ型 char(255) として保存されます。

例

例 1 CONTEXT1 という名前のアプリケーション・コンテキストを作成し、それに属性 ATTR1 とその値 VALUE1 を設定します。

```
select set_appcontext("CONTEXT1", "ATTR1", "VALUE1")
---------------
0
```

例 2 既存のアプリケーション・コンテキストの上書きを試みます。試みは失敗し、-1 が返されます。

```
select set_appcontext("CONTEXT1", "ATTR1", "VALUE1")
--------------
-1
```

例 3 set_appcontext に値のデータ型の変換を組み込む方法を示します。

```
declare@val numeric
select @val = 20
select set_appcontext ("CONTEXT1", "ATTR2",
convert(char(20), @val))
------------
0
```

例 4　適切なパーミッションを持たないユーザがアプリケーション・コンテキストを設定しようとしたときの結果を示します。試みは失敗し、-1 が返されます。

```
select set_appcontext("CONTEXT1", "ATTR2", "VALUE1")
--------------
-1
```

使用法

- set_appcontext は、成功すると 0 を返し、失敗すると -1 を返します。
- 現在のセッションに既に存在する値を設定すると、set_appcontext は -1 を返します。
- set_appcontext では、既存のアプリケーション・コンテキストの値は上書きできません。コンテキストに新しい値を割り当てるには、コンテキストを削除してから、新しい値を使用して再作成してください。
- set_appcontext は、属性を char データ型として格納します。作成するアクセス・ルールで属性値を別のデータ型と比較する必要がある場合は、アクセス・ルールで char データを適切なデータ型に変換する必要があります。
- この関数では、すべての引数が必須です。

get_appcontext

指定したコンテキストの属性値を返します。

get_appcontext ("*context_name*", "*attribute_name*")

パラメータ

- *context_name* − アプリケーション・コンテキスト名を指定するロー。データ型 char(30) として保存されます。
- *attribute_name* − アプリケーション・コンテキスト属性名を指定するロー。データ型 char(30) として保存されます。

例

例 1　ATTR1 の値として VALUE1 が返されています。

```
select get_appcontext ("CONTEXT1", "ATTR1")
-----------
VALUE1
```

例 2　ATTR1 は CONTEXT2 にはありません。

```
select get_appcontext("CONTEXT2", "ATTR1")
------------
NULL
```

例 3　適切なパーミッションを持たないユーザがアプリケーション・コンテキストを取得しようとしたときの結果を示します。

```
select get_appcontext("CONTEXT1", "ATTR2")
select permisssion denied on built-in get_appcontext, database
dbid
----------
-1
```

使用法

- get_appcontext は、成功すると 0 を返し、失敗すると -1 を返します。
- 指定された属性がアプリケーション・コンテキスト内にない場合は、get_appcontext は “null” を返します。
- get_appcontext は、属性を char データ型として格納します。作成するアクセス・ルールで、属性値を他のデータ型と比較する場合は、アクセス・ルールで char データを適切なデータ型に変換する必要があります。
- この関数では、すべての引数が必須です。

list_appcontext

現在のセッション内にある全コンテキストの属性をすべてリストします。

list_appcontext ("*context_name*")

パラメータ

- *context_name* – セッション内のアプリケーション・コンテキスト属性をすべて指定します。list_appcontext のデータ型は char(30) です。

例

例 1 適切なパーミッションを持つユーザがアプリケーション・コンテキストの一覧を表示したときの結果を示します。

```
select list_appcontext ("*", "*")
Context Name: (CONTEXT1)
Attribute Name: (ATTR1) Value: (VALUE2)
Context Name: (CONTEXT2)
Attribute Name: (ATTR1) Value: (VALUE!)
-----------
0
```

例 2 適切なパーミッションを持たないユーザがアプリケーション・コンテキストの一覧を表示しようとしたときの結果を示します。試みは失敗し、-1 が返されます。

```
select list_appcontext()
Select permission denied on built-in
list_appcontext, database DBID
---------
-1
```

使用法

- list_appcontext は、成功すると 0 を返し、失敗すると -1 を返します。
- 組み込み関数が複数の結果セットを返すことはないので、クライアント・アプリケーションは list_appcontext の戻り値をメッセージとして受け取ります。

パーミッション

list_appcontext を使用するには、ユーザに適切なパーミッションが付与されている必要があります。詳細については、「アプリケーション・コンテキスト関数を使ってパーミッションを設定する」(218 ページ) を参照してください。

rm_appcontext

特定のアプリケーション・コンテキストまたはすべてのアプリケーション・コンテキストを削除します。

rm_appcontext ("*context_name*", "*attribute_name*")

パラメータ

- *context_name* − アプリケーション・コンテキスト名を指定するロー。データ型 char(30) として保存されます。
- *attribute_name* − アプリケーション・コンテキスト属性名を指定するロー。データ型 char(30) として保存されます。

例

例 1　アスタリスク ("*") を使用して、指定したコンテキスト内のすべての属性を削除します。

```
select rm_appcontext("CONTEXT1", "*")
---------
0
```

例 2　アスタリスク ("*") を使用して、すべてのコンテキストと属性を削除します。

```
select rm_appcontext("*", "*")
---------
0
```

例 3　存在しないコンテキストを削除しようとしています。試みは失敗し、-1 が返されます。

```
select rm_appcontext("NON_EXISTING_CTX", "ATTR2")
---------
-1
```

例 4　適切なパーミッションを持たないユーザがアプリケーション・コンテキストを削除しようとしたときの結果を示します。

```
select rm_appcontext("CONTEXT1", "ATTR2")
---------
-1
```

使用法

- rm_appcontext は、成功すると 0 を返し、失敗すると -1 を返します。
- この関数では、すべての引数が必須です。

SYS_SESSION システム・アプリケーション・コンテキスト

SYS_SESSION コンテキストを使用すると、デフォルトの事前定義アプリケーション・コンテキストが表示されます。これには、セッション固有の属性と値のペアが定義されています。このコンテキストを使用する構文は次のとおりです。

```
select list_appcontext ("SYS_SESSION", "*")
```

その後で、次の構文を使用します。

```
select get_appcontext ("SYS_SESSION", "<attribute>")
```

表 6-4: SYS_SESSION の属性と値

属性	値
username	ログイン名
hostname	クライアントの接続元ホスト名
applname	クライアントによって設定されたアプリケーション名
suserid	現在のデータベースでのユーザのユーザ ID
groupid	現在のデータベースでのユーザのグループ ID
dbid	ユーザの現在のデータベースの ID
dbname	現在のデータベース
spid	サーバ・プロセス ID
proxy_suserid	代理のサーバ・ユーザ ID
client_name	set client_name コマンドを使用して中間層アプリケーションによって設定されたクライアント名
client_applname	set client_applname コマンドを使用して中間層アプリケーションによって設定されたクライアント・アプリケーション名
client_hostname	set client_hostname コマンドを使用して中間層アプリケーションによって設定されたクライアント・ホスト名
language	デフォルトの、または set language で設定された、クライアントが現在使用している言語 (@@language)
character_set	クライアントが使用している文字セット (@@client_csname)
dateformat	set dateformat を使用して設定された、クライアントが受け取る日付の形式
is_showplan_on	set showplan がオンの場合は YES、オフの場合は NO
is_noexec_on	no exec がオンの場合は YES、オフの場合は NO

アクセス・ルールと ACF による問題の解決

この項では、ある問題の解決方法を示します。その問題とは、セキュリティ・レベルが異なる 5 人のユーザが、それぞれのユーザのセキュリティ・レベル以下の値を持つローだけを参照できるようにするというものです。この解決方法では、アクセス・ルールを Application Context Facility とともに使用し、Dave というユーザが参照するローだけを表示します。

次の 5 つのログインがあります。

- Anne のセキュリティ・レベルは 1 です。
- Bob のセキュリティ・レベルは 1 です。
- Cassie のセキュリティ・レベルは 2 です。
- Dave のセキュリティ・レベルは 2 です。
- Ellie のセキュリティ・レベルは 4 です。

各ユーザが参照できるローは、rlac の値が自分のセキュリティ・レベル以下であるローだけとなるようにする必要があります。このようにするには、アクセス・ルールを作成して ACF を適用します。

rlac カラムは integer 型、appcontext 引数は char 型です。

```
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)
```

ログイン・トリガの使用

注意 この項の情報の一部は、「Login triggers in ASE 12.5 (ASE 12.5)」(http://www.sypron.nl/logtrig.html) (Copyright 1998–2002, Rob Verschoor/ Sypron B.V) からの引用です。

ログイン・トリガは、ユーザがログインするたびに、指定されたストアド・プロシージャを実行します。ログイン・トリガは、バックグラウンドで実行される点を除けば、通常のストアド・プロシージャと同じです。これは、正常なログイン・プロセスの最後のステップとして実行され、ログインするユーザのアプリケーション・コンテキストを設定します。

サーバ内のユーザに対してログイン・トリガを登録できるのは、システム・セキュリティ担当者だけです。

安全な環境を実現するには、システム管理者は次のことを実行する必要があります。

1 set_appcontext 関数に対する select 権限を取り消します。ログイン・トリガの所有者は、sa_role を付与されたユーザであっても、set_appcontext を使用するには明示的なパーミッションが必要です。

2 ストアド・プロシージャからログイン・トリガを設定し、そのログイン・トリガをユーザに登録します。

3 ユーザが実行するログイン・トリガに実行権限を設定します。

ログイン・トリガの作成

ログイン・トリガは、ストアド・プロシージャとして作成します。create trigger コマンドは使用しないでください。次の例では、まず、pubs2 データベースに lookup テーブルを作成する必要があります。

```
create table lookup (
  appname varchar(20),
  attr varchar(20),
  value varchar(20),
  login varchar(20)
)
```

次に、pubs2 データベース内にログイン・トリガのストアド・プロシージャを作成します。

```
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
```

loginproc の実行パーミッションを public に付与します。

```
grant execute on loginproc to public
```

特定のユーザにログイン・トリガを関連付けるには、そのユーザのデフォルト・データベースで alter login を実行します。

ログイン・トリガの設定

ログイン・トリガを設定、変更、または削除するには、有効な sso_role が必要です。ログイン・トリガのオブジェクト ID は、syslogins.procid カラムに保存されます。デフォルトでは、ログイン・トリガは存在しません。ログイン・トリガは、alter login を使用して登録する必要があります。

このコマンドは、ユーザのデフォルト・データベースで実行します。ログイン・トリガとして登録するストアド・プロシージャは、ユーザのデフォルト・データベース内になければなりません。Adaptive Server はユーザのデフォルト・データベースの sysobjects テーブルでログイン・トリガ・オブジェクトを検索するからです。

ログイン・トリガの設定

次の例では、Adaptive Server ログイン my_login のログイン・トリガとしてストアド・プロシージャ my_proc (設定するデータベース内に存在している必要があります) を設定します。

```
alter login my_login modify login script "my_proc"
```

この場合も、コマンドはユーザのデフォルト・データベースから実行する必要があります。Adaptive Server では、このストアド・プロシージャに対する execute 権限がログインにあるかどうかの検査が行われますが、ユーザが実際にログインしてログイン・トリガを実行するまでは権限の検査は行われません。

ログイン・トリガの削除と変更

ログイン・トリガとして設定されているストアド・プロシージャを削除することはできません。初めに設定を解除する必要がありますが、それにはログイン・トリガを完全に削除するか、ログイン・トリガの設定を別のストアド・プロシージャに変更します。ログイン・トリガを削除するには、次のように入力します。

```
alter login my_login drop login script
```

ログイン・トリガの設定を別のストアド・プロシージャに変更するには、次のように入力します。

```
alter login my_login modify login script "diff_proc"
```

ログイン・トリガの表示

現在のログイン・トリガを表示するには、sp_displaylogin を使用します。

```
sp_displaylogin my_login
go
(....)
Default Database: my_db
Default Language:
Auto Login Script: my_proc
....
```

ログイン・トリガの実行

ログイン・トリガが通常のストアド・プロシージャと異なるのは、登録されたログイン・トリガは、アクティブなユーザ接続を持たずにバックグラウンドで実行される点です。ログイン・トリガが設定されている場合は、そのユーザがログインすると、Adaptive Server はクライアント・アプリケーションからの何らかのコマンドを実行する前にログイン・トリガをバックグラウンドで自動的に実行します。

1 つのログインで複数の同時接続を確立する場合は、ログイン・トリガはセッションごとに独立して実行されます。同様に、複数のログインが同じストアド・プロシージャをログイン・トリガとして設定することもできます。

ログイン・トリガとして設定されたストアド・プロシージャはバックグラウンドで実行されるので、ストアド・プロシージャの標準機能の中には使用できなくなるものがあります。たとえば、デフォルト値のないパラメータをプロシージャとの間で受け渡すことはできません。また、プロシージャが結果の値を返すことはありません。

この特別な実行モードは、ログイン・トリガのストアド・プロシージャによって呼び出されるすべてのプロシージャと、ログイン・トリガのストアド・プロシージャ自体によって生成されるすべての出力に影響を与えます。

ログイン・トリガのストアド・プロシージャを通常のストアド・プロシージャとして、たとえば、isql から実行することもできます。プロシージャは通常どおりに動作し、出力とエラー・メッセージもすべて通常どおり表示されます。

ログイン・トリガの出力について

ストアド・プロシージャをバックグラウンド・タスクとして実行した場合の最大の影響は、一部のエラー・メッセージと同様に、ログイン・トリガからの出力がクライアント・アプリケーションではなく Adaptive Server エラー・ログ・ファイルに書き込まれることです。

エラー・ログでは、print または raiserror のメッセージの出力は background task message または background task error というテキストで始まります。たとえば、ログイン・トリガ内で print “Hello!” という文と raiserror 123456 という文を実行した場合は、Adaptive Server エラー・ログには次のように出力されます。

```
(....) background task message: Hello!
(....) background task error 123456: This is test
message 123456
```

ただし、すべての出力が Adaptive Server エラー・ログに書き込まれるわけではありません。

- select 文の結果セットは、通常であればクライアント接続に送信されますが、この場合は Adaptive Server エラー・ログも含めてどこにも出力されません。この情報は消滅します。
- 正常に実行される文には、insert...select 文と select...into 文の他に、通常は結果セットをクライアント・アプリケーションに送信しないその他の DML 文、および通常のストアド・プロシージャ内で実行可能な DDL 文があります。

その他のアプリケーションでのログイン・トリガの使用

ログイン・トリガは、Adaptive Server のロー・レベル・アクセス制御機能の一部です。したがって、セッションが Adaptive Server にログインした後は、ログイン・トリガをアクセス・ルールおよびアプリケーション・コンテキストと組み合わせて使用することにより、ロー・レベル・アクセス制御を設定することができます。ただし、ログイン・トリガは他の目的で使用することもできます。

同時接続数の制限

次の例では、1 つのログインで確立できる Adaptive Server への同時接続数を制限します。この例の手順 1 と 2 で説明する各コマンドは、アクセス制限の対象となるユーザのデフォルト・データベースで実行されます。

1　システム管理者として、limit_user_sessions ストアド・プロシージャを次のように作成します。

```
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 システム・セキュリティ担当者として、このストアド・プロシージャをユーザ “bob” のログイン・トリガとして設定します。

```
alter login bob modify login script
"limit_user_sessions"
go
```

3 ユーザ “bob” が Adaptive Server の 3 番目のセッションを作成するとき、syb_quit() 関数を呼び出すログイン・トリガによってこのセッションを終了します。

```
% 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 このメッセージは、Adaptive Server のエラー・ログ・ファイルに記録されます。

```
(...) background task message: Aborting login
[ my_login]: exceeds session limit [2]
```

時間ベースの制限の適用

次の例では、システム管理者がログイン・トリガを作成して、ユーザ・セッションに対して時間ベースの制限を適用する方法を示します。手順 1 ～ 4 で説明する各コマンドは、アクセス制限の対象となるユーザのデフォルト・データベースで実行されます。

1 システム管理者として次のテーブルを作成します。

```
create table access_times (
suid int not null,
dayofweek tinyint,
shiftstart time,
shiftend time)
```

2　システム管理者として、テーブル access_times に次のようなローを挿入します。これらのローでは、ユーザ “bob” は、月曜日の午前 9 時～午後 5 時に Adaptive Server へのログインを許可され、ユーザ “mark” は、火曜日の午前 9 時～午後 5 時に Adaptive Server へのログインを許可されることが示されています。

```
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　システム管理者として limit_access_time ストアド・プロシージャを作成します。このストアド・プロシージャでは、access_time テーブルを参照して、ログイン・アクセスを許可するかどうかを決定します。

```
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 システム・セキュリティ担当者として、limit_access_time ストアド・プロシージャをユーザ “bob” とユーザ “mark” のログイン・トリガとして設定します。

```
alter login bob login script
"limit_access_time"
go
alter login mark login script
"limit_access_time"
go
```

5 月曜日に、ユーザ “bob” はセッションを正常に作成できます。

```
isql -Ubob -Ppassword
1> select 1
2> go
-----------
          1
(1 row affected)
```

しかし、ユーザ “mark” の Adaptive Server へのアクセスは拒否されます。

```
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 次のメッセージがエラー・ログに書き込まれます。

```
(...) server back-ground task message: Aborting
login [mark]: login attempt past normal working
hours
```

上記の例では、特定のログインの同時接続数を制限し、このログインのアクセスを特定の時間帯だけに制限しました。ただし、欠点が 1 つあります。それは、セッションが終了した理由をクライアント・アプリケーションが容易に検出できないことです。ユーザに、たとえば「ユーザ数が多すぎます。後でやり直してください」などのメッセージを表示するには、別の方法を使用します。

現在のセッションを終了させるだけの組み込み関数 syb_quit() を呼び出す代わりに、ストアド・プロシージャ内でエラーを発生させて、ログイン・トリガのストアド・プロシージャをアボートします。

たとえば、ゼロ除算を行うとログイン・トリガのストアド・プロシージャがアボートし、セッションが終了して、メッセージが表示されます。

ログイン・トリガの制限事項

次のアクションは制限を受けます。

- #temp テーブルを作成して後でそのセッション内で使用することはできません。他のストアド・プロシージャの場合と同様に、プロシージャが完了すると #temp テーブルは自動的に削除され、元のセッション設定がリストアされます。
- sa ログインにはログイン・トリガを使用しないでください。ログイン・トリガが失敗すると、Adaptive Server からロック・アウトされる場合があります。
- 数秒以上かかるような処理をログイン・トリガで実行すると処理の問題が生じる場合があるので、そのような処理にはログイン・トリガを使用しないでください。

問題と情報

- Adaptive Server エラー・ログにアクセスできない場合は、ログイン・トリガを使用しないでください。常に Adaptive Server エラー・ログでエラー・メッセージを確認してください。
- Adaptive Server バージョン 15.0.2 以降では、ログイン・トリガでエクスポート可能なオプションを設定または解除すると、サーバが起動する時点のログイン・プロセスで反映されます。

  この動作を無効にするには、ログイン・トリガ内で set export_options off を実行します。

  Adaptive Server バージョン 15.0.1、12.5.4、およびそれ以前では、ログイン・トリガのオプションを有効にするには、トレース・フラグ 4073 を有効にして Adaptive Server を起動する必要があります。
- isql などのクライアント・アプリケーションは、ログイン・トリガの存在や実行を認識しません。ログインに成功すると、すぐにクライアント・アプリケーションのコマンド・プロンプトが表示されますが、Adaptive Server によってコマンドが実行されるのはログイン・トリガが正常に実行された後です。この isql のプロンプトは、ログイン・トリガによってユーザ接続が終了した場合でも表示されます。
- Adaptive Server にログインするユーザには、ログイン・トリガのストアド・プロシージャを使用するための execute パーミッションが必要です。execute パーミッションが付与されていない場合は、Adaptive Server のエラー・ログにエラー・メッセージが出力され、ユーザ接続はただちに終了します (ただし、isql のコマンド・プロンプトは表示されます)。

  Adaptive Server のエラー・ログには、次のようなメッセージが出力されます。

```
EXECUTE permission denied on object my_proc,
database my_db, owner dbo
```

- ログイン・トリガのストアド・プロシージャのパラメータには、必ずデフォルト値を設定してください。ストアド・プロシージャのパラメータの中にデフォルト値がないものが見つかると、ログイン・トリガは失敗し、Adaptive Server のエラー・ログに次のようなエラーが出力されます。

  ```
  Procedure my_proc expects parameter @param1, which
  was not supplied...
  ```

ログイン・トリガに対する実行権限の無効化

データベース所有者または管理者は、ログイン・トリガに対する execute 権限を無効化することができます。あるいは、特定の場合にのみアクセスを許可するようにログイン・トリガをコーディングすることもできます。たとえば、データベース所有者または管理者がテーブルを更新している間は、一般のユーザがサーバを使用できないようにする場合です。

注意 ログイン・トリガが負の数を返した場合は、ログインは失敗です。

ログイン・トリガからの set オプションのエクスポート

Adaptive Server では、ログイン・トリガ内の set コマンドのオプションをユーザ・セッション全体で有効にできます。

次の set オプションは自動的にエクスポートされます。

- 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

グローバル・ログイン・トリガの設定

グローバル・ログイン・トリガを設定するには、sp_logintrigger を使用します。これは、ユーザのログインごとに実行されます。ユーザ固有のアクションを取得するには、alter login もしくは create login を使用してユーザ固有のログイン・トリガを設定します。

注意 トレース・フラグ -T4073 を設定して、このオプションをアクティブ化できます。

第 7 章

データの機密保持

この章では、すべてのデータを保護し、機密性を保持するための Adaptive Server の設定方法について説明します。

Adaptive Server における SSL (Secure Sockets Layer)

Adaptive Server Enterprise セキュリティ・サービスは、現在 SSL (Secure Sockets Layer) セッションベースのセキュリティをサポートしています。**SSL** は、クレジット・カード番号、株式売買、銀行取引などの機密情報を、インターネット上で安全に転送するための標準です。

このマニュアルでは、パブリック・キー暗号法については詳しく説明しませんが、SSL によってインターネット通信チャネルの安全性が保証される仕組みを理解できるように、基本的なことについては説明します。このマニュアルは、パブリック・キー暗号法の全般的なガイドではありません。

Adaptive Server SSL 機能の実装は、ユーザ・サイトのセキュリティ・ポリシーとニーズを熟知し、SSL およびパブリック・キー暗号法について全般的な知識のあるシステム・セキュリティ担当者がいることを前提としています。

インターネット通信の概要

TCP/IP は、クライアント／サーバ・コンピューティングで使用されるプライマリ・トランスポート・プロトコルであり、インターネットへのデータ転送を制御するプロトコルです。TCP/IP では、送信側から受信側へデータが転送されるときに、いくつもの中間コンピュータを経由します。複数のコンピュータを経由することによって、通信システムの中に安全性の低いリンクが生じ、データの改ざん、盗難、盗聴、なりすましなどを受けやすくなります。

パブリック・キー暗号法

「パブリック・キー暗号法」とは、機密を要するデータをインターネットでの転送中に保護するために開発され、実装されている、さまざまなメカニズムの総称です。パブリック・キー暗号法は、暗号化、キー交換、デジタル署名、デジタル証明書から構成されます。

復号化

暗号化のプロセスでは、暗号化アルゴリズムを使用して情報をコード化し、その情報を目的の受信者以外の者から保護します。暗号化に使用するキーには、次の 2 種類があります。

- **対称キー暗号化**では、メッセージの暗号化と復号化に同じアルゴリズム (キー) を使用します。この暗号化方式では、簡単に解読できる単純なキーを使用しているため、最低限のセキュリティしか保証されません。しかし、対称キーによる暗号化の場合は、メッセージの暗号化と復号化に必要な計算の量が最小限で済むため、データ転送が高速になります。
- **パブリック・キー／プライベート・キー** (非対称キー) 暗号化では、公開コンポーネントと秘密コンポーネントから成る 1 対のキーを使用してメッセージの暗号化と復号化を行います。通常、送信者はプライベート・キーを使用してメッセージを暗号化し、受信者は送信者のパブリック・キーを使用してメッセージを復号化しますが、この組み合わせは異なる場合もあります。送信者が受信者のパブリック・キーを使ってメッセージを暗号化し、受信者が受信者自身のプライベート・キーを使用してメッセージを復号化することも可能です。

  パブリック・キーとプライベート・キーを作成するときに使用するアルゴリズムは複雑なので、解読するのは容易ではありません。しかし、パブリック・キー／プライベート・キー暗号化では、より多くの計算が必要となり、接続を介して送られるデータの量も増えるので、データ転送が遅くなります。

キー交換

安全性を損なうことなく、計算によるオーバヘッドを減らしてトランザクションを高速化するには、対称キー暗号化とパブリック・キー／プライベート・キー暗号化の両方を組み合わせて使用します。この方法を、キー交換と呼びます。

データ量が多い場合は、対称キーを使用して元のメッセージを暗号化します。次に、送信者は、送信者自身のプライベート・キーまたは受信者のパブリック・キーを使用して、対称キーを暗号化します。暗号化されたメッセージと暗号化された対称キーの両方が受信者に送信されます。メッセージを暗号化するときにはパブリック・キーまたはプライベート・キーを使用しますが、そのときに使用しなかった方のキーを使用して、受信者は対称キーを復号化します。キーの交換が終了すると、受信者は対称キーを使用してメッセージを復号化します。

デジタル署名

デジタル署名は、不正な変更を検出したり拒否を禁止したりする場合に使用されます。テキスト／メッセージからユニークな固定長の文字列になった数字を生成する数値アルゴリズムを使用して、デジタル署名は作成されます。この生成された数値はハッシュまたはメッセージ・ダイジェストと呼ばれます。

メッセージの整合性を保証するために、メッセージ・ダイジェストは署名者のプライベート・キーで暗号化され、ハッシュ・アルゴリズムについての情報とともに受信者に送信されます。受信者は、署名者のパブリック・キーを使用してメッセージを復号化します。また、この処理では、元のメッセージ・ダイジェストも再生成されます。これらのダイジェストが一致すれば、メッセージは損なわれておらず、改ざんされてもいないことになります。一致しない場合は、転送中にデータが修正されたか、改ざん者によりデータが署名されたことになります。

さらに、デジタル署名によって「否認防止」が可能になります。つまり、送信者は、自身のプライベート・キーでメッセージを暗号化するので、メッセージを送ったことを否定 (否認) できないことになります。ただし、盗難や解読によってプライベート・キーの機密性が損なわれると、デジタル署名は否認防止に役立ちません。

デジタル証明書

デジタル証明書はパスポートのようなものです。証明書がユーザに割り当てられると、認証局は、システムにおけるユーザのあらゆる ID 情報を持つことになります。パスポートと同様に、証明書は、あるエンティティ (サーバ、ルータ、Web サイトなど) の身元を他者に対して確認するために使用されます。

Adaptive Server は次の 2 つのタイプの証明書を使用します。

- **サーバ証明書** − サーバ証明書は、それを保有しているサーバを認証します。証明書は、信頼された第三者の CA (認証局) によって発行されます。CA は、証明書の保有者の身元を検証し、保有者のパブリック・キーなどの ID 情報を、デジタル証明書に埋め込みます。証明書には、発行元 CA のデジタル署名が含まれています。これによって、証明書データの整合性が確認され、証明書を使用できるようになります。
- **認証局証明書 (信頼されたルート証明書**とも呼ばれます) − サーバの起動時にロードされる、信頼された認証局のリストです。認証局証明書は、RPC (リモート・プロシージャ・コール) の間などサーバがクライアントとして機能するときに、サーバによって使用されます。Adaptive Server は、自身の認証局の信頼されたルート証明書を起動時にロードします。Adaptive Server は、RPC を実行するためにリモート・サーバに接続するときに、リモート・サーバの証明書に署名した CA が、Adaptive Server 自身の CA の信頼されたルート・ファイルにある「信頼された」CA かどうかを検証します。信頼された CA でない場合は、接続が許可されません。

証明書は一定期間有効で、認証局は、セキュリティ侵害が生じたときなどさまざまな理由で証明書を無効にすることができます。セッション中に証明書が無効になった場合、そのセッション接続は継続します。後続のログイン試行は失敗します。同様に、証明書の有効期限が切れたときも、ログイン試行は失敗します。

これらのメカニズムの組み合わせにより、インターネットを介して送信されるデータを盗聴や改ざんから守ります。また、なりすまし攻撃からもユーザを保護します。なりすまし攻撃には、あるエンティティが別のエンティティの振りをする (スプーフィング) ものや、組織または個人が、機密情報の入手という本当の目的を隠して別の目的を偽るもの (虚偽の陳述) があります。

SSL の概要

SSL は、ワイヤ・レベルまたはソケット・レベルで暗号化されたデータを、保護されたネットワーク接続を介して送信するための業界標準です。

サーバとクライアントは何度か I/O を交換し、安全な暗号化セッションをネゴシエートして合意してから、SSL 接続が確立されます。これは、SSL ハンドシェイクと呼ばれています。

SSL ハンドシェイク

クライアントが接続を要求すると、SSL が有効化されているサーバは、その身元を証明する証明書を提示してから、データ転送を行います。基本的に、ハンドシェイクは次の手順から成り立っています。

- クライアントはサーバに接続要求を送信します。要求には、クライアントがサポートしている SSL (または TLS: Transport Layer Security) オプションが含まれています。
- サーバは、自身の証明書と、サポートされている暗号スイートのリストを返す。このリストには、SSL/TLS サポート・オプション、キー交換で使用するアルゴリズム、デジタル署名が含まれます。
- クライアントとサーバがお互いに CipherSuite に合意すると、安全で暗号化されたセッションが確立されます。

SSL ハンドシェイクと SSL/TLS プロトコルの詳細については、Internet Engineering Task Force Web サイト (http://www.ietf.org) を参照してください。

Adaptive Server がサポートする暗号スイートのリストについては、「暗号スイート」(249 ページ) を参照してください。

Adaptive Server での SSL

Adaptive Server が SSL を実装したことにより、いくつかのレベルでのセキュリティが可能になりました。

- サーバが自身を認証し (ユーザの交信対象のサーバであることを証明する)、データ転送を行う前に、暗号化された SSL セッションを開始する。
- SSL セッションが確立すると、接続を要求するクライアントは暗号化された安全な接続を介してユーザ名とパスワードを送信できる。
- サーバ証明書の電子署名を比較することにより、クライアントが受信したデータが、本来の受信者に到達するまでに修正されたかどうかを判断できる。

ほとんどのプラットフォームで、Adaptive Server は Certicom の SSL Plus(TM) ライブラリ API を使用しています。ただし、Windows Opteron X64 では、Adaptive Server は SSL プロバイダとして OpenSSL を使用しています。

SSL フィルタ

interfaces ファイル、Windows レジストリ、LDAP サービスなどの Adaptive Server のディレクトリ・サービスは、サーバ・アドレスとポート番号を定義し、クライアント接続に使用するセキュリティ・プロトコルを決定します。Adaptive Server では、SSL プロトコルはフィルタとして実装され、ディレクトリ・サービスの master 行と query 行に追加されます。

Adaptive Server が接続を受け付けるアドレスとポート番号は、単一のサーバで複数のネットワーク・プロトコルとセキュリティ・プロトコルを有効にできるように設定することが可能です。サーバ接続の属性は、LDAP などのディレクトリ・サービス、または従来の Sybase の *interfaces* ファイルで指定されます。「サーバ・ディレクトリ・エントリの作成」(246 ページ) を参照してください。

SSL フィルタを使用して *interfaces* ファイルの master エントリまたは query エントリに接続するには、その接続で SSL プロトコルをサポートしている必要があります。SSL 接続を受け付け、別の接続では暗号化されないクリア・テキストを受け付けるようにサーバを設定することも、他のセキュリティ・メカニズムを使用するように設定することもできます。

たとえば、SSL ベースの接続とクリア・テキストの接続の両方をサポートする UNIX の *interfaces* ファイルは、次のようになります。

```
SYBSRV1
master tcp ether myhostname myport1 ssl
query    tcp ether myhostname myport1 ssl
master tcp ether myhostname myport2
```

SSL フィルタは、*interfaces* ファイル (Windows では *sql.ini*) の SECMECH (セキュリティ・メカニズム) 行で定義される Kerberos などのセキュリティ・メカニズムとは別のものです。

証明書による認証

SSL プロトコルは、暗号化セッションを有効にするために、サーバ証明書によるサーバ認証を要求します。同様に、Adaptive Server が RPC の実行時にクライアントとして機能しているときには、サーバ証明書を検証するためにクライアント接続がアクセスできる、信頼された認証局のレポジトリが必要になります。

サーバ証明書

それぞれの Adaptive Server には、起動時にロードされる専用のサーバ証明書ファイルが必要です。証明書ファイルのデフォルトのロケーションは次のとおりです。*servername* は、起動時にコマンド・ラインで-s フラグを使用して、または環境変数 *$DSLISTEN* を使用して指定される Adaptive Server の名前です。

- UNIX － *$SYBASE/$SYBASE_ASE/certificates/servername.crt*
- Windows － *%SYBASE%\%SYBASE_ASE%\certificates\servername.crt*

サーバ証明書ファイルは、サーバ証明書と、そのサーバ証明書用の暗号化されたプライベート・キーを含む、コード化されたデータから構成されています。

また、sp_ssladmin を使用して、サーバ証明書ファイルのロケーションを指定することもできます。

注意 クライアントが正しく接続できるようにするには、証明書内の共通名が *interfaces* ファイル内の Adaptive Server 名と一致している必要があります。

認証局の信頼されたルート証明書

信頼された認証局のリストは、Adaptive Server の起動時に、信頼されたルート・ファイルからロードされます。信頼されたルート・ファイルは、フォーマットは証明書ファイルに似ていますが、Adaptive Server が認識する認証局の証明書が格納されている点が異なります。信頼されたルート・ファイルは次のロケーションにあり、ローカルの Adaptive Server からアクセスできます。*servername* はサーバ名です。

- UNIX − *$SYBASE/$SYBASE_ASE/certificates/servername.txt*
- Windows − *%SYBASE%\%SYBASE_ASE\certificates\servername.txt*

信頼されたルート・ファイルが使用されるのは、RPC や CIS (コンポーネント総合サービス) 接続の実行時など、Adaptive Server がクライアントとして機能しているときだけです。

Adaptive Server が受け付ける認証局をシステム・セキュリティ担当者が追加および削除するには、一般的な ASCII テキスト・エディタを使用します。

警告！ Adaptive Server 内部では、システム・セキュリティ担当者の役割 (sso_role) を使用して、セキュリティに関係するオブジェクトに対するアクセスや実行を制限してください。

Adaptive Server には、証明書要求を生成するツールや証明書を認可するためのツールがあります。「Adaptive Server ツールを使用した証明書の要求と認可」(245 ページ) を参照してください。

接続タイプ

この項では、クライアントとサーバの間のさまざまな接続について説明します。

クライアントから Adaptive Server へのログイン

既存のクライアント接続が確立されるのと同じように、Open Client アプリケーションは Adaptive Server へのソケット接続を確立します。ネットワーク・トランスポート・レベルの接続コールがクライアント側で完了し、承認コールがサーバ側で完了すると、ソケット上で SSL ハンドシェイクが行われ、その後でユーザ・データが送信されます。

サーバ間リモート・プロシージャ・コール (RPC)

Adaptive Server が RPC を実行するために他のサーバへのソケット接続を確立する方法は、既存の RPC 接続の確立方法と同じです。ネットワーク・トランスポート・レベルの接続コールが完了して、ソケット上で SSL ハンドシェイクが行われた後で、ユーザ・データが送信されます。サーバ間のソケット接続が既に確立している場合は、既存のソケット接続とセキュリティ・コンテキストが再使用されます。

Adaptive Server は、RPC の実行時にクライアントとして機能しているときは、接続中にリモート・サーバの証明書を要求します。Adaptive Server は、リモート・サーバの証明書に署名した認証局が信頼できることを確認します。つまり、信頼されたルート・ファイルにある、自身の信頼された認証局のリストにあることを確認します。また、サーバ証明書内の共通名が、接続の確立時に使用した共通名と一致していることを確認します。

コンパニオン・サーバと SSL

コンパニオン・サーバを使用してフェールオーバを行うように Adaptive Server を設定できます。プライマリ・サーバとセカンダリ・サーバの両方で、SSL と RPC の設定が同じであるように設定してください。接続がフェールオーバまたはフェールバックされるとき、接続とともにセキュリティ・セッションが再度確立されます。

Open Client 接続

コンポーネント統合サービス、RepAgent、分散トランザクション管理、および Adaptive Server の他のモジュールは、Client Library を使用して Adaptive Server 以外のサーバとの接続を確立します。リモート・サーバはその証明書によって認証されます。リモート・サーバは、ユーザ名とパスワードを使用して、RPC を実行するための Adaptive Server クライアント接続を認証します。

SSL の有効化

Adaptive Server は、interface ファイル (Windows では *sql.ini*) に基づいて、各ポートで使用するセキュリティ・サービスを判断します。

❖ SSL の有効化

1　サーバの証明書を生成します。

2　信頼されたルート・ファイルを作成します。

3　sp_configure を使用して、SSL を有効にします。コマンド・プロンプトで、次のように入力してください。

```
sp_configure "enable ssl", 1
```

- 1 − 起動時に SSL サブシステムが有効になり、メモリが割り当てられます。ネットワーク上で送受信されるデータは SSL によってワイヤレベルで暗号化されます。
- 0 (デフォルト) − SSL を無効にします。これはデフォルトの設定です。

4　SSL フィルタを *interfaces* ファイルに追加します。「サーバ・ディレクトリ・エントリの作成」(246 ページ) を参照してください。

5 sp_ssladmin を使用して、証明書ファイルに証明書を追加します。「証明書の管理」(247 ページ) を参照してください。

6 Adaptive Server を停止して再起動します。

注意 第三者の証明書を要求、認証、変換するには、『ユーティリティ・ガイド』のcertauth、certreq、certpk12 の各ツールの説明を参照してください。

Kerberos、NTLAN などの他のセキュリティ・サービスとは異なり、SSL は、Open Client/Open Server 設定ファイル *libtcl.cfg* の “Security” セクションにも、*objectid.dat* 内のオブジェクトにも依存しません。

システム管理者は、物理メモリの総量を計画するときに、SSL で使用するメモリを考慮する必要があります。Adaptive Server で SSL 接続を行う場合、接続ごとに約 40K のメモリが必要になります (接続にはユーザ接続、リモート・サーバ、ネットワーク・リスナを含む)。メモリは、メモリ・プール内で予約され事前に割り付けられ、Adaptive Server ライブラリと SSL Plus ライブラリにより必要に応じて内部で使用されます。

証明書の取得

システム・セキュリティ担当者は、次の手順で、Adaptive Server のサーバ証明書とプライベート・キーをインストールします。

- ユーザ環境に導入されている既存のパブリック・キー・インフラストラクチャ (PKI) に用意されているサードパーティのツールを使用する。
- 信頼された第三者の認証局と Adaptive Server 証明書要求ツールを組み合わせて使用する。

証明書を取得するには、認証局 (CA) に証明書を要求してください。Adaptive Server では、PEM フォーマットを使用するための SSL 証明書が必要です。しかし、認証局から PEM 以外のフォーマットの認証が交付される場合があります。第三者に要求した証明書が PKCS #12 フォーマットである場合は、certpk12 ユーティリティを使用して、その証明書を Adaptive Server で認識できるフォーマットに変換してください (『ユーティリティ・ガイド』を参照)。

Adaptive Server 証明書要求ツールをテストし、その認証方法がサーバで機能していることを確認するために、Adaptive Server では、ユーザが認証局として機能し、認証局が署名した証明書をユーザ自身に発行できるようにするツールをテスト用に用意しています。

Adaptive Server で使用する証明書を作成するには、次の手順に従います。

1 パブリック・キーとプライベート・キーのペアを生成します。

2 プライベート・キーを安全な場所に保管します。

3 証明書要求を生成します。

4 証明書要求を CA に送信します。

5　認証局が署名した証明書が返されたら、その証明書をファイルに保存し、プライベート・キーを証明書に追加します。

6　Adaptive Server インストール・ディレクトリに証明書を格納します。

証明書を要求するサードパーティ・ツール

ほとんどのサードパーティ PKI ベンダといくつかのブラウザには、証明書とプライベート・キーを生成するユーティリティがあります。これらのユーティリティの多くはグラフィカルなウィザード形式で、一連の質問にユーザが答えると証明書の識別名と共通名が定義されます。

ウィザードの指示に従って、証明書要求を作成します。署名済みの PKCS #12 フォーマット証明書を受け取ったら、certpk12 を使用して、証明書ファイルとプライベート・キー・ファイルを生成します。この 2 つのファイルを連結して *servername.crt* ファイルを作成します。*servername* はサーバ名です。このファイルは、$SYBASE/$SYBASE_ASE の下の *certificates* ディレクトリに置いてください。詳細については、『ユーティリティ・ガイド』を参照してください。『ASE ユーティリティ・ガイド』を参照してください。

Adaptive Server ツールを使用した証明書の要求と認可

Adaptive Server には、証明書の要求と認証を行う 2 つのツールがあります。certreq は、パブリック・キーとプライベート・キーのペアと証明書要求を生成します。certauth は、サーバ証明書要求を認証局署名済み証明書に変換します。

警告！ certauth は、テスト専用で使用します。商用 CA のサービスを利用することをおすすめします。こうしたサービスではルート証明書の整合性が保護されており、広く承認された CA により署名された証明書を使用すれば、クライアント証明書を使用する形式の認証への移行が促進されるからです。

サーバの信頼されたルート証明書を用意するには、5 つの手順を実行します。最初の 2 つの手順では、テスト版の信頼されたルート証明書を作成します。ここで、サーバ証明書を作成できることを確認できます。検査用の CA 証明書 (信頼されたルート証明書) を作成したら、3 ～ 5 の手順を繰り返してサーバ証明書に署名します。

1　certreq を使用して、証明書を要求します。

2　certauth を使用して、証明書要求を認証局自己署名証明書 (信頼されたルート証明書) に変換します。

3　certreq を使用して、サーバ証明書とプライベート・キーを要求します。

4　certauth を使用して、証明書要求を認証局署名済みサーバ証明書に変換します。

5 プライベート・キーのテキストをサーバ証明書に付加して、サーバのインストール・ディレクトリに証明書を格納します。

注意 Adaptive Server では openssl オープン・ソース・ユーティリティが *$SYBASE/$SYBASE_OCS/bin* に含まれています。certreq、certauth、certpk12 で実装されたすべての証明書管理タスクを実行するには openssl を使用します。Sybase では便宜上このバイナリを組み込んでいますが、バイナリを使用して発生した問題についてはいっさい責任を負いません。詳細については、www.openssl.org を参照してください。

第三者証明書の要求、認証、変換に使用する Sybase ユーティリティ certauth、certreq、certpk12 の詳細については、『ユーティリティ・ガイド』を参照してください。

注意 certauth と certreq は、RSA と DSA のアルゴリズムに依存しています。これらのツールは、RSA アルゴリズムおよび DSA アルゴリズムを使用して証明書要求を構築する暗号モジュールと組み合わせる場合にのみ機能します。

サーバ・ディレクトリ・エントリの作成

Adaptive Server は、クライアント・ログインとサーバ間の RPC を受け入れます。Adaptive Server が接続を受け入れるアドレスやポート番号は設定可能であり、複数のネットワーク、さまざまなプロトコル、代替ポートを指定することができます。

interfaces ファイルでは、SSL は master 行と query 行でのフィルタとして指定しますが、Kerberos などのセキュリティ・メカニズムを指定するには SECMECH 行を使用します。以下の例は、UNIX 環境で Adaptive Server に SSL を使用する場合の TLI ベース・エントリを示しています。

Windows で SSL および Kerberos セキュリティ・メカニズムを使用するサーバのエントリは、次のように設定します。

```
[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
```

例における SYBSRV2 の SECMECH 行には、Kerberos のセキュリティ・メカニズムをそれぞれ参照する OID (オブジェクト識別子) が含まれています。OID の値は次のファイルに定義されています。

- UNIX － *$SYBASE/$SYBASE_OCS/config/objectid.dat*
- Windows － *%SYBASE%\%SYBASE_OCS\ini\objectid.dat*

これらの例では、SSL セキュリティ・サービスはポート番号 2748 (0x0abc) に設定されています。

注意 SSL を SECMECH セキュリティ・メカニズムと同時に使用する意図は、SECMECH から SSL セキュリティへのマイグレーションを容易にすることにあります。

証明書の管理

Adaptive Server で SSL や証明書を管理するには、sp_ssladmin を使用します。このストアド・プロシージャを実行するには sso_role が必要です。

sp_ssladmin では次のことを実行できます。

- ローカル・サーバ証明書を追加する。証明書を追加して、プライベート・キーの暗号化に使用するパスワードを指定することも、起動時にコマンド・ラインからのパスワード入力を要求するようにすることもできる。
- ローカル・サーバ証明書を削除する。
- サーバ証明書のリストを表示する。

sp_ssladmin の構文は次のとおりです。

```
sp_ssladmin {[addcert, certificate_path [, password|NULL]]
      [dropcert, certificate_path]
      [lscert]
      [help]}
      [lsciphers]
      [setciphers, {"FIPS" | "Strong" | "Weak" | "All"
      | quoted_list_of_ciphersuites}]
```

次に例を示します。

```
sp_ssladmin addcert, "/sybase/ASE-12_5/certificates/Server1.crt",
      "mypassword"
```

この設定により、ローカル・サーバの証明書ファイル *Server1.crt* を、絶対パス */sybase/ASE-12_5/certificates* (Windows の場合は *x:\sybase\ASE-12_5\certificates*) に追加します。プライベート・キーは、パスワード *mypassword* を使用して暗号化されています。プライベート・キーの作成時に指定したパスワードを指定してください。

証明書を受け入れる前に、sp_ssladmin は次のことを確認します。

- 指定されたパスワードを使用してプライベート・キーを復号化できる (NULL が指定された場合を除く)。
- 証明書のプライベート・キーとパブリック・キーが一致する。
- ルート認証局からサーバ証明書までの証明書チェーンが正しい。
- 証明書内の共通名が、*interfaces* ファイル内の共通名と一致する。

共通名が一致しない場合は、sp_ssladmin は警告を発行します。その他の基準が満たされない場合は、その証明書は証明書ファイルに追加されません。

警告！ Adaptive Server では、パスワードは最大 64 文字です。さらに、プラットフォームによっては、サーバ証明書の作成時に有効なパスワード長が制限されます。次の制限の範囲内でパスワードを選択してください。

- Sun Solaris － 32 ビットおよび 64 ビットの両方のプラットフォーム、最大 256 文字
- Linux － 128 文字
- IBM － 32 ビットおよび 64 ビットの両方のプラットフォーム、32 文字
- HP － 32 ビットおよび 64 ビットの両方のプラットフォーム、8 文字
- Windows － 256 文字

NULL をパスワードとして使用する意図は、SSL 暗号化セッションを開始する前の、SSL の初期設定の間パスワードを保護することにあります。SSL はまだ設定されていないので、パスワードは暗号化されずに接続を介して送られます。最初のログイン時にパスワードを NULL に指定すると、これを防止できます。

NULL をパスワードにした場合は、-y フラグを付けて dataserver を開始する必要があります。このとき、コマンド・ラインでプライベート・キーのパスワードを入力するためのプロンプトが表示されます。

SSL 接続が確立された状態で Adaptive Server を再起動した後、実際のパスワードを使用して sp_ssladmin を再実行します。このパスワードは暗号化されて保管されます。その後、コマンド・ラインから Adaptive Server を起動するときは、この暗号化されたパスワードが使用されるので、管理者が起動時にコマンド・ラインからパスワードを指定する必要がなくなります。

最初のログイン時に NULL のパスワードを使用する方法の代わりに、isql を使用した Adaptive Server へのリモート接続をできないようにするという方法があります。*interfaces* ファイル (Windows では *sql.ini*) 内の *hostname* として “localhost” を指定すると、クライアントはリモート接続できなくなります。ローカル接続だけが確立できるので、パスワードがネットワーク接続を介して転送されることはありません。

注意 Adaptive Server のネットワーク・メモリ・プールには十分なメモリがあるため、sp_ssladmin addcert はデフォルトのメモリ割り付けを使用して証明書とプライベート・キーを設定できます。ただし、ネットワーク・メモリを消費する別のプログラムがデフォルト・ネットワーク・メモリの割り付けを既に行っていた場合、sp_ssladmin は失敗し、次のエラーがクライアントに対して出力されます。

```
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)
```

または、次のメッセージがログ・ファイルに書き込まれます。

```
 ... ssl_alloc: Cannot allocate using ubfalloc(rnetmempool,
131072)
```

対処方法として、additional network memory 設定パラメータの値を大きくすることができます。sp_ssladmin addcert が正常に終了するには約 500K バイトのメモリが必要なので、additional network memory をこの値まで大きくすることで、操作を成功させることができます。このメモリは、必要に応じてネットワーク・メモリ・プールで再使用されます。または、sp_ssladmin が正常に完了した後、additional network memory の値を元に戻すこともできます。

パフォーマンス

安全なセッションの確立に必要な、追加のオーバヘッドがあります。データを暗号化するとサイズが増え、情報の暗号化と復号化に追加の計算が必要になるからです。SSL の追加メモリ要件は、ネットワーク・スループットまたは接続を確立するためのオーバヘッドを 50 ～ 60 パーセント増加させます。ユーザ接続ごとに約 40K のメモリがさらに必要になります。

暗号スイート

SSL ハンドシェイク中に、クライアントとサーバは CipherSuite を介して共通のセキュリティ・プロトコルをネゴシエートします。暗号スイートは、SSL 対応のアプリケーションで使用されるキー交換アルゴリズム、ハッシュ方式、暗号化方式の優先順位付きリストです。暗号スイートの詳細については、IETF (Internet Engineering Task Force) の Web ページ (http://www.ietf.org/rfc/rfc2246.txt) を参照してください。

デフォルトでは、クライアントとサーバの両方がサポートしている最強の CipherSuite が SSL ベースのセッションに使用されます。

Adaptive Server は、SSL Plus ライブラリ API と暗号エンジンである Security Builder™ (両方とも Certicom 製) で使用可能な暗号スイートをサポートしています。

注意 上記にリストした暗号スイートは、TLS (トランスポート・レイヤ仕様) に準拠しています。TLS は SSL 3.0 を拡張したものであり、SSL バージョン 3.0 暗号スイートの別名です。

@@ssl_ciphersuite

Transact-SQL グローバル変数 *@@ssl_ciphersuite* によって、ユーザは SSL ハンドシェイクでどの暗号スイートが選択されたか、また、SSL または非 SSL 接続が確立されているか知ることができます。

Adaptive Server は、SSL ハンドシェイクが完了したときに *@@ssl_ciphersuite* を設定します。値は、非 SSL 接続であることを示す NULL、または SSL ハンドシェイクで選択された暗号スイートの名前を含む文字列のいずれかになります。

たとえば、SSL プロトコルを使用する isql 接続では、この接続で選択された暗号スイートが表示されます。

```
1>  select @@ssl_ciphersuite
2>  go
```

出力：

```
------------------------------
TLS_RSA_WITH_AES_128_CBC_SHA

(1 row affected)
```

SSL 暗号スイートの優先度の設定

Adaptive Server の sp_ssladmin には、暗号スイートの優先度を表示および設定するためのコマンド・オプションとして、lsciphers と setciphers という 2 つのコマンド・オプションがあります。これらのオプションによって Adaptive Server が使用する暗号スイートのセットを制限することで、システム・セキュリティ担当者はサーバに対するクライアント接続や Adaptive Server からのアウトバウンド接続で使われる暗号化アルゴリズムの種類をコントロールすることができます。Adaptive Server で SSL 暗号スイートを使用する場合のデフォルトの動作は以前のバージョンと変わりません。暗号スイートのために内部的に定義された優先度セットが使われます。

暗号スイートの優先度セットの値を表示するには、次のように入力します。

```
sp_ssladmin lsciphers
```

特定の暗号スイートの優先度を設定するには次のように入力します。

```
sp_ssladmin setciphers, {"FIPS" | "Strong" | "Weak" | "All" |
quoted_list_of_ciphersuites }
```

各パラメータの意味は、次のとおりです。

- “FIPS” － FIPS に準拠した暗号化、ハッシュ、キー交換アルゴリズムのセット。このリストに含まれるアルゴリズムは AES、3DES、DES、SHA1 です。
- “Strong” － 64 ビットより長いキーを使用する暗号化アルゴリズムのセット。
- “Weak” － サポート対象のすべての暗号スイートのセットの中で強力セットのカテゴリーに含まれない暗号化アルゴリズムのセット。
- “All” － デフォルトの暗号スイートのセット。
- quoted_list_of_ciphersuites － 暗号スイートのセットを、優先度順にカンマで区切ったリストで指定します。引用符 (“) でリストの先頭と最後をマークします。引用符で囲んだリストに、個々の暗号スイート名のほか、定義済みの任意のセットを含めることができます。未知の暗号スイート名を指定するとエラーが報告され、優先度は変更されません。

定義済みのセットの詳細な内容については、表 7-1 (252 ページ) を参照してください。

sp_ssladmin setciphers は、指定された順序リストに暗号スイートの優先度を設定します。これは使用可能な SSL 暗号スイートを、“FIPS”、“Strong”、“Weak”、“All”、または引用符で囲まれた暗号スイート・リストのセットに制限します。これが有効になるのは次のリスナが開始されたときで、Adaptive Server を再起動してすべてのリスナが新しい設定を使うようにする必要があります。

設定されている任意の暗号スイートの優先度を、sp_ssladmin lsciphers で表示することができます。優先度が設定されていない場合、sp_ssladmin lsciphers は 0 個のローを返します。これは優先度が設定されておらず、Adaptive Server がデフォルトの (内部) 優先度を使用することを意味します。

表 7-1: Adaptive Server の定義済み暗号スイート

セット名	セット内の暗号スイート名
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

セット名	セット内の暗号スイート名
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 は、Adaptive Server 15.0 以降ではサポートされない暗号スイートを示します。削除された暗号スイートを使用すると SSLHandshake が失敗し、Adaptive Server には接続できません。

表 7-2: 削除された暗号スイート

セット名	セットから削除された暗号スイート名
FIPS	TLS_RSA_EXPORT1024_WITH_DES_CBC_SHA
Strong	削除なし
Weak	TLS_RSA_EXPORT1024_WITH_RC4_56_SHA
	TLS_DHE_DSS_EXPORT_WITH_DES40_CBC_SHA
	TLS_DHE_RSA_EXPORT_WITH_DES40_CBC_SHA
その他のデッドロック	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 の例

最初に開始されるときは、まだ暗号スイートの優先度が設定されていないので、sp_ssladmin lscipher は優先度を表示しません。

```
1> sp_ssladmin lscipher
2> go
```

出力：

```
 Cipher Suite Name   Preference
-----------------    ----------
(0 rows affected)
(return status = 0)
```

次の例では、FIPS アルゴリズムを使用する暗号スイートのセットを指定しています。

```
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
```

優先度 0 (ゼロ) の sp_ssladmin 出力は、Adaptive Server で使用されない暗号スイートを示します。他のゼロ以外の値は、SSL ハンドシェイクの間に Adaptive Server がアルゴリズムを使用する優先度の順序を示します。SSL ハンドシェイクのクライアント側はこれらの暗号スイートから、受け付ける暗号スイートのリストに一致するものを選びます。

この例では、引用符で囲んだ暗号スイートのリストで、Adaptive Server に優先度を設定しています。

```
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
```

その他の注意事項

Adaptive Server バージョン 12.5.3 以降にアップグレードしたときは、サーバのデフォルトが暗号スイートの優先度になり、sp_ssladmin のオプション lscipher で優先度は表示されません。サーバはそのデフォルトの優先度、すなわち “All” で定義される優先度を使用します。システム・セキュリティ担当者は、自分のサイトのセキュリティ・ポリシーと使用可能な SSL 暗号スイートを検討し、暗号スイートを制限するかどうかや、どの暗号スイートがセキュリティ・ポリシーに合っているかを判断する必要があります。

Adaptive Server バージョン 12.5.3 以降からアップグレードするときに、暗号スイートの優先度が設定されている場合、設定された優先度がアップグレード後も使用されます。アップグレードの完了後に、サーバの暗号スイートの優先度が現在のセキュリティ・ポリシーに適合していることを確認し、表 7-1 の暗号スイート・リストで暗号スイートがサポートされているかどうかを調べてください。サポートされていない暗号スイートは削除してください。

設定した SSL 暗号スイートの優先度をサーバからすべて削除してデフォルトの優先度を使いたい場合は、次のコマンドを使用してシステム・カタログ内の記憶領域から優先度を削除します。

```
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
```

これらのコマンドは、システム・セキュリティ担当者またはシステム管理者だけが実行できます。

SSL を使用した共通名の指定

ディレクトリ・サービス・エントリで指定したサーバ名は、SSL ハンドシェイクを実行する際に SSL サーバ証明書が使用する共通名とは異なる可能性があります。これにより、SSL 証明書の共通名の完全修飾ドメイン名 (たとえば、*server1.bigcompany.com*) を使用できます。

interfaces ファイルに共通名を追加するには、次のコマンドを使用します。

```
ase1
   master tcp ether host_name port_number ssl="CN='common_name'"
   query tcp ether host_name port_number ssl="CN='common_name'"
```

SSL を使用する Adaptive Server にクライアントが SSL を使用して接続する場合は、*interfaces* ファイルのポート番号の後に SSL フィルタが配置されます。ディレクトリ・サービスには、dsedit またはテキスト・エディタを使用して追加できる共通名が含まれます。

sp_listener での共通名の指定

sp_listener にはパラメータ CN=*common_name* が含まれており、SSL 証明書の共通名を指定できます。構文は次のとおりです。

```
sp_listener 'command','[protocol:]machine_name:port_number:
"CN=common_name"', 'engine_number'
```

プロトコルとして ssltcp を指定する場合にのみ、CN=*common_name* を使用します。ここで指定する *common_name* は SSL 証明書の *common_name* に照らして検証されます。CN=*common_name* を含めない場合、Adaptive Server は *server_name* を使用して SSL 証明書の共通名に照らして検証します。証明書に完全修飾ドメイン名を含める場合、このドメイン名は CN=*common_name* と一致する必要があります。

属性名 “CN” は大文字と小文字を区別しません (“CN”、“cn”、または “Cn” を使用できます) が、共通名の属性値は大文字と小文字を区別します。

たとえば、共通名 `ase1.big server 1.com` を指定するには、次のように入力します。

```
sp_listener 'start','ssltcp:blade1:17251:"CN=ase1.big server 1.com"','0'
```

sp_listener の詳細については、『リファレンス・マニュアル：プロシージャ』を参照してください。

変更されたストアド・プロシージャ *sp_addserver*

filter パラメータは、共通名を指定するように拡張されています。『リファレンス・マニュアル：プロシージャ』を参照してください。

Kerberos による機密保持

Adaptive Server では、すべてのメッセージの機密性を保持することもできます。Adaptive Server との間で送受信するすべてのメッセージが暗号化されることを要求するには、msg confidentiality reqd 設定パラメータを 1 に設定します。このパラメータが 0 (デフォルト) の場合、メッセージの機密保持は要求されませんが、機密保持を行うかどうかをクライアント側で設定することは可能です。

たとえば、すべてのメッセージを暗号化するように要求するには、次のコマンドを実行します。

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

Kerberos やサポートされているその他のセキュリティ・サービスを使用したメッセージの機密保持の詳細については、「ネットワークベース・セキュリティの管理」(92 ページ) を参照してください。

パスワード保護を使用したデータベースのダンプとロード

dump database コマンドの password パラメータを使用すると、権限を持たないユーザがデータベース・ダンプをロードできないように保護することができます。データベース・ダンプの作成時に password パラメータを指定した場合には、データベースのロード時にもこのパスワードを指定する必要があります。

パスワード保護に対応する dump database コマンドと load database コマンドの構文の一部を次に示します。

dump database *database_name* to *file_name* [with passwd = *password*]

load database *database_name* from *file_name* [with passwd = *password*]

各パラメータの意味は、次のとおりです。

- *database_name* − ダンプまたはロードするデータベースの名前です。
- *file_name* − ダンプ・ファイルの名前です。
- *password* − 不正なユーザからダンプ・ファイルを保護するために指定するパスワードです。

パスワードの長さは、6 ～ 30 文字にする必要があります。6 文字より短く 30 文字より長いパスワードを指定すると、Adaptive Server からエラー・メッセージが発行されます。データベースをロードするときに誤ったパスワードを発行すると、Adaptive Server からエラー・メッセージが発行され、コマンドは失敗します。

たとえば、次の例はパスワード “bluesky” を使用して pubs2 データベースのデータベース・ダンプを保護します。

```
dump database pubs2 to "/Syb_backup/mydb.db" with passwd = "bluesky"
```

このデータベース・ダンプをロードするときには同じパスワードを使用する必要があります。

```
load database pubs2 from "/Syb_backup/mydb.db" with passwd = "bluesky"
```

パスワードと以前のバージョンの Adaptive Server

パスワード保護に対応する dump コマンドと load コマンドを使用できるのは、Adaptive Server バージョン 12.5.2 以降のみです。Adaptive Server バージョン 12.5.2 のダンプに password パラメータを使用した場合、そのダンプを以前のバージョンの Adaptive Server にロードしようとすると失敗します。

パスワードと文字セット

ダンプをロードできるサーバは同じ文字セットを使用しているサーバのみです。たとえば、ASCII 文字セットを使用するサーバから ASCII 以外の文字セットを使用するサーバにダンプをロードしようとすると、ASCII のパスワードの値は ASCII ではないパスワードと異なるためロードが失敗します。

ユーザが入力したパスワードは、Adaptive Server のローカル文字セットに変換されます。ASCII 文字は通常は文字セット間で値の表現が同じであるため、ユーザのパスワードが ASCII 文字セットであれば、dump と load のパスワードはすべての文字セットで認識されます。

Adaptive Server バージョン 15.0.2 以降では、ポータブル・パスワードを保存できます。「パスワード文字セットの考慮事項」(40 ページ) を参照してください。

第 8 章

監査

この章では、インストール環境に応じた監査の設定方法について説明します。

Adaptive Server での監査の概要

安全なシステムを構築するうえで重要な要素は、責任の所在を明確にすることです。この責任を確実に保つ手段の 1 つとして、システムのイベントを監査する方法があります。Adaptive Server で発生する多くのイベントは記録が可能です。

監査は、データベース管理システムのセキュリティの重要な機能です。監査証跡を使用して、システムへの侵入とリソースの不正使用を検出します。システム・セキュリティ担当者は、監査証跡を調べることによって、データベース内のオブジェクトに対するアクセスのパターンを調べて特定のユーザのアクティビティをモニタできます。監査レコードを追跡すればユーザを特定できるので、システムを不正に使用しようとするユーザに対する抑止力となります。

各監査レコードには、イベントの性質、日時、イベントの責任者、イベントが正常か失敗かについて記録できます。監査できるイベントには、ログインとログアウト、サーバの起動、データ・アクセス・コマンドの使用、特定オブジェクトへのアクセス、特定ユーザのアクションなどがあります。「監査証跡」(監査レコードのログ) によって、システム・セキュリティ担当者はシステムで発生したイベントを再構築し、イベントの影響を判断できます。

システム・セキュリティ担当者は、監査の開始と停止、監査オプションの設定、監査データの処理を行うことができる唯一のユーザです。システム・セキュリティ担当者は、次のようなイベントの監査を設定できます。

- サーバ全体にわたるセキュリティ関連イベント
- データベース・オブジェクトの作成、削除、変更

- 特定ユーザが行ったすべてのアクション、または特定の役割をアクティブにしてユーザが行ったすべてのアクション
- データベース・アクセス権の付与または取り消し
- データのインポートまたはエクスポート
- ログインとログアウト

Adaptive Server とオペレーティング・システムの監査レコードの関連付け

Adaptive Server の監査レコードをオペレーティング・システムの監査レコードにリンクするには、Adaptive Server のログイン名をオペレーティング・システムのログイン名と同じにするのが最も簡単です。

あるいは、システム・セキュリティ担当者が、ユーザのオペレーティング・システム・ログイン名をそのユーザの Adaptive Server ログイン名にマッピングすることもできます。ただし、この方法では、新規ユーザのログイン名を手作業で登録しなければならず、運用中の保守が必要となります。

監査システム

監査システムは、次のものからなります。

- グローバル監査オプションと監査証跡を含む sybsecurity データベース
- 監査証跡に書き込まれる前の監査レコードが格納される、メモリ内の監査キュー
- 監査を管理するための設定パラメータ
- 監査を管理するためのシステム・プロシージャ

sybsecurity データベース

sybsecurity データベースは、監査機能のインストール・プロセス中に作成されます。model データベース内のすべてのシステム・テーブルの他に、このデータベースには、サーバ全体の監査オプション追跡用のシステム・テーブル sysauditoptions と監査証跡用のシステム・テーブルが含まれます。

sysauditoptions の内容は、グローバル監査オプションの現在の設定値です。これは、ディスク・コマンド、リモート・プロシージャ・コール、独自のユーザ定義監査レコード、またはすべてのセキュリティ関連イベントに対する監査を有効にするかどうかなどを設定するものです。これらのオプションは Adaptive Server 全体に影響します。

監査証跡

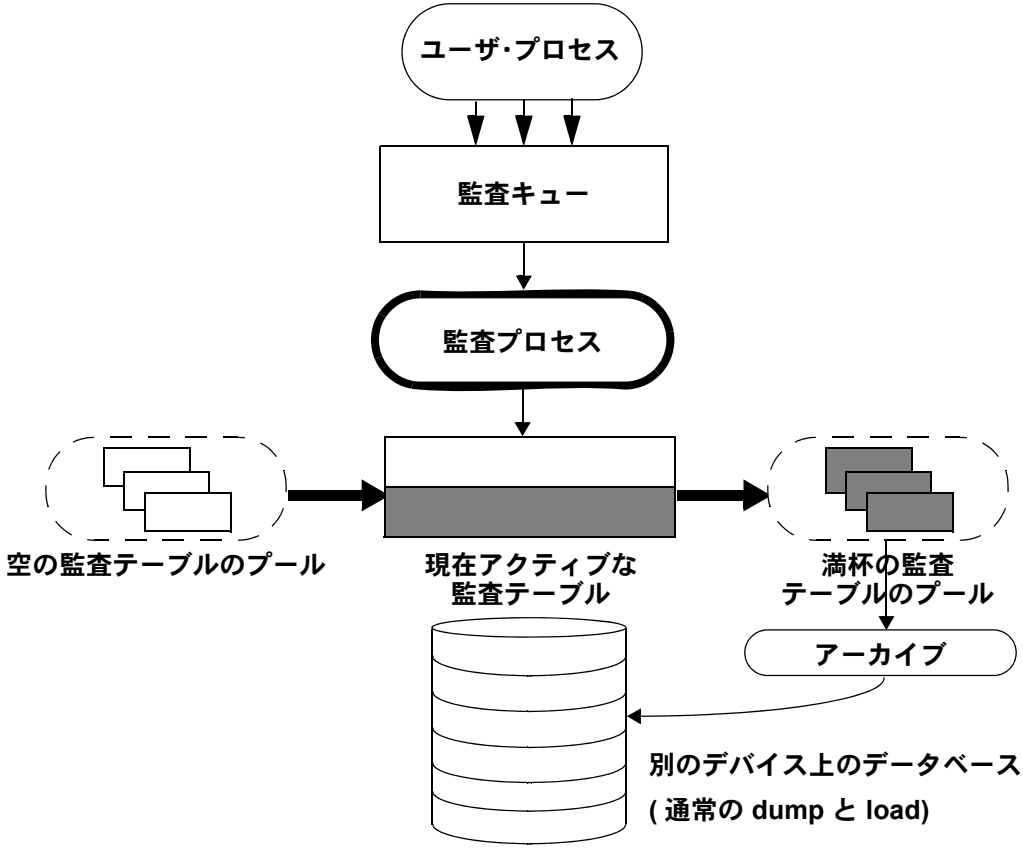
Adaptive Server は、sysaudits_01 から sysaudits_08 までのシステム・テーブルに監査証跡を格納します。監査機能をインストールするときに、インストール環境に合わせた監査テーブルの数を決定します。たとえば、2 つの監査テーブルを使用する場合は、その名前は sysaudits_01 と sysaudits_02 となります。「現在の監査テーブル」は、常に 1 つしかありません。Adaptive Server は、現在の監査テーブルにすべての監査データを書き込みます。システム・セキュリティ担当者は sp_configure を使用して、どの監査テーブルを現在のものにするかを設定したり、変更したりできます。

監査テーブル数は 2 以上とし、各テーブルを個別の監査デバイス上に置くことをおすすめします。このようにすれば、監査レコードが失われることや手動介入を必要とすることなく、監査テーブルがアーカイブされ、処理されるので、監査プロセスはスムーズに実行されるようになります。

警告！ Sybase では、運用システムに対して単一の監査テーブルを使用しないよう強くおすすめします。使用する監査テーブルが 1 つだけの場合は、監査レコードが失われるおそれがあります。システム・リソースの制約から単一の監査テーブルしか使用できない場合は、「単一テーブル監査」(276 ページ) の指示を参照してください。

図 8-1 は、監査プロセスが複数の監査テーブルをどのように処理するかを示しています。

図 *8-1:* 複数の監査テーブルを使用した監査



監査システムは、メモリ内監査キューから現在の監査テーブルに監査レコードを書き込みます。現在の監査テーブルが満杯に近づいたときに、スレッショルド・プロシージャによってそのテーブルを自動的に別のデータベースにアーカイブできます。アーカイブ・データベースは、dump コマンドと load コマンドによってバックアップおよびリストアできます。バックアップからアーカイブされた監査テーブルに対して読み込み専用アクセスを行うには、アーカイブ・データベースへのアクセスを使用します。『システム管理ガイド 第 2 巻』の「第 14 章 アーカイブ・データベースへのアクセス」を参照してください。監査証跡の管理の詳細については、「監査証跡の管理の設定」(268 ページ) を参照してください。

監査キュー

監査イベントが発生すると、監査レコードは、まずメモリ内の監査キューに格納されます。このレコードは、監査プロセスによって監査証跡に書き込まれるまで、メモリ内に残ります。監査キューのサイズは、sp_configure の audit queue size パラメータを使用して設定できます。

監査キューのサイズを設定するにあたっては、システム・クラッシュ時にキュー内のレコードが失われる危険性と、キューが満杯になったときのパフォーマンスのロスとのトレードオフについて考慮してください。監査レコードがキュー内にある限り、システム・クラッシュによってレコードが失われる可能性はあります。しかし、キューが頻繁に満杯になるようでは、システム全体のパフォーマンスに影響します。ユーザ・プロセスが監査レコードを生成しようとしたときに監査キューに空きがない場合は、そのプロセスは、キュー内のスペースが使用可能になるまでスリープします。

注意 監査レコードは監査証跡に直接書き込まれるのではないので、監査レコードが現在の監査テーブルにすぐに保管されるとは考えないでください。

監査設定パラメータ

監査プロセスの管理には、次の設定パラメータを使用します。

- auditing は、Adaptive Server 全体の監査を有効または無効にします。このパラメータは、sp_configure の実行後すぐに反映されます。このパラメータが有効な場合にのみ監査が実行されます。
- audit queue size は、監査キューのサイズを設定します。このパラメータは、メモリの割り付けに影響を与えるため、Adaptive Server が再起動されるまでは有効になりません。
- suspend audit when device full は、監査デバイスが満杯になったときの監査プロセスの動作を制御します。このパラメータは、sp_configure の実行後すぐに反映されます。
- current audit table は、現在の監査テーブルを設定します。このパラメータは、sp_configure の実行後すぐに反映されます。

監査用のシステム・プロシージャ

監査プロセスの管理には、次のシステム・プロシージャを使用します。

- sp_audit は、監査オプションを有効または無効にします。監査対象のイベントを指定するのに必要なシステム・プロシージャはこれだけです。
- sp_displayaudit は、アクティブな監査オプションを表示します。

- sp_addauditrecord は、監査証跡にユーザ定義監査レコード (コメント) を追加します。ユーザがこの種のレコードを追加できるのは、システム・セキュリティ担当者が sp_audit を使って独自の監査を有効にした場合のみです。

監査のインストールと設定

表 8-1: 監査の一般的な手順

アクションと説明	参照箇所
1. 監査のインストール − 監査テーブル数を設定する。監査証跡および sybsecurity データベース内の syslogs トランザクション・ログにデバイスを割り当てる。	「監査システムのインストール」(264 ページ) と、Adaptive Server の『インストール・ガイド』および『設定ガイド』を参照。
2. 監査証跡の管理の設定 − 現在の監査テーブルがほとんど満杯になったときに制御を受け取るスレッショルド・プロシージャを作成して設定する。このプロシージャは、自動的に新しい監査テーブルに切り替えて、現在のテーブルの内容をアーカイブする。 また、この手順では、audit queue size と suspend audit when device full の各設定パラメータも設定する。	「監査証跡の管理の設定」(268 ページ) 単一テーブルでの監査については、「単一テーブル監査」(276 ページ) を参照。
3. sybsecurity データベース内のトランザクション・ログの管理の設定 − sybsecurity データベースの syslogs トランザクション・ログの処理方法と trunc log on chkpt データベース・オプションの設定方法を決定し、trunc log on chkpt がオフの場合の syslogs に対するラストチャンス・スレッショルド・プロシージャを構築する。	「トランザクション・ログの管理の準備」(274 ページ)
4. 監査オプションの設定 − sp_audit を使用して、監査対象のイベントを設定する。	「グローバル監査オプションの設定」(280 ページ)
5. 監査の有効化 − sp_configure を使用して auditing 設定パラメータをオンにする。Adaptive Server は、現在の監査テーブルへの監査レコードの書き込みを開始する。	「監査の有効化と無効化」(276 ページ)
6. 監査の再起動 − 監査が失敗した場合は、sp_audit restart を使用して監査を再起動する。	「監査の再起動」(279 ページ)

監査システムのインストール

監査システムは、通常、Sybase インストール・プログラムの auditinit を使用してインストールします。あるいは、auditinit を使用しないで監査システムをインストールすることもできます。詳細については、「installsecurity による監査のインストール」(265 ページ) を参照してください。インストールと auditinit については、プラットフォームの『Adaptive Server インストール・ガイド』および『Adaptive Server 設定ガイド』を参照してください。

監査機能をインストールするときに、監査証跡に使用するシステム・テーブルの数、各監査システム・テーブル用のデバイス、syslogs トランザクション・ログ用のデバイスを設定できます。

監査証跡のためのテーブルとデバイス

指定できるシステム・テーブルは最高 8 つです (sysaudits_01 から sysaudits_08 まで)。監査証跡には、少なくとも 2 つのテーブルを使用するようにしてください。各テーブルは、マスタ・デバイスとは別に独自のデバイスに配置します。このようにすると、スレッショルド・プロシージャを使用して、現在の監査テーブルが満杯になる前にその内容を自動的にアーカイブしてから、新しい空のテーブルに切り替えてそれ以降の監査レコードを保存することができます。

syslogs トランザクション・ログ・テーブルのデバイス

監査機能をインストールするときに、syslogs システム・テーブルで構成されるトランザクション・ログ用に個別のデバイスを指定する必要があります。この syslogs テーブルは、すべてのデータベースに存在するもので、そのデータベースで実行されるトランザクションのログが格納されます。

installsecurity による監査のインストール

$SYBASE/ASE-15_0/scripts ディレクトリに、監査機能をインストールするためのスクリプトである *installsecurity* があります。

注意　この例では、サーバが使用する論理ページ・サイズは 2K であるとします。

installsecurity を使用して監査機能をインストールするには、次の手順に従います。

1　disk init コマンドと create database コマンドを使用して、監査デバイスと監査データベースを作成します。次に例を示します。

```
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　isql を使用して、*installsecurity* スクリプトを実行します。

```
cd $SYBASE/ASE-12_5/scripts
setenv DSQUERY server_name
isql -Usa -Ppassword -Sserver_name < installsecurity
```

3　Adaptive Server を停止して再起動します。

これらの手順を終了すると、sybsecurity データベースの独自セグメントに 1 つの監査テーブル (sysaudits_01) が作成されます。この時点で監査を有効にすることは可能ですが、sp_addaudittable システム・プロシージャを使用して、さらに監査テーブルを追加する必要があります。disk init、create database、sp_addaudittable の詳細については、『リファレンス・マニュアル：プロシージャ』を参照してください。

複数デバイスへの監査データベースの移動

sybsecurity データベースは、master データベースとは別の独自のデバイス上に置く必要があります。複数の監査テーブルがある場合は、テーブルをそれぞれ専用のデバイスに配置します。各テーブルを別のセグメントに置き、それぞれが別のデバイスを指すようにすると便利です。現在 master と同じデバイスに sybsecurity がある場合、または何らかの理由で別のデバイスに sybsecurity を移動したい場合は、以下の項で説明する手順のいずれかを使用してください。データベースを移動する場合、既存のグローバル監査設定を保存するかどうかを指定できます。

グローバル監査設定を保存しない *sybsecurity* の移動

注意 この手順にはsybsecurity データベースの削除が含まれます。この削除によって、sybsecurity で記録されている監査レコードとグローバル監査設定がすべて破棄されます。sybsecurity データベースを削除する前に、必ず、バックアップによって、または「監査テーブルのアーカイブ」(269 ページ) の手順に従って既存のレコードをアーカイブし、sybsecurity テーブルに残っている履歴データを保管してください。

グローバル監査設定を保存しないで、sybsecurity データベースを移動するには、次の手順に従います。

1 次のコマンドを実行し、ログインに関連する情報を syslogins システム・テーブルから削除します。

```
sp_audit "all","all","all","off"
```

2 sybsecurity データベースを削除します。

3 次で説明しているいずれかのインストール手順に従い、sybsecurity をもう一度インストールします。

- 使用しているプラットフォームの設定ガイド
- 「installsecurity による監査のインストール」(265 ページ).

4 このインストール・プロセスで、sybsecurity データベースを、必ずマスタ・デバイスとは別の 1 つまたは複数のデバイス上に置くようにしてください。

sybsecurity の移動とグローバル監査設定の保存

❖ ***sybsecurity* データベースを移動して、グローバル監査設定を保存するには、次の手順に従います。**

1 sybsecurity データベースをダンプします。

```
dump database sybsecurity to "/remote/sec_file"
```

2 sybsecurity データベースを削除します。

```
drop database sybsecurity
```

3 sybsecurity データベースを配置する最初のデバイスを初期化します。

```
disk init name = "auditdev",
    physname = "/dev/dsk/c2d0s4",
    size = "10M"
```

4 セキュリティ・ログを配置するデバイスを初期化します。

```
disk init name = "auditlogdev",
    physname = "/dev/dsk/c2d0s5",
    size = "2M"
```

5 新しい sybsecurity データベースを作成します。

```
create database sybsecurity on auditdev
    log on auditlogdev
```

6 古い sybsecurity データベースの内容を、新しく作成したデータベースにロードします。グローバル監査設定は維持されます。

```
load database sybsecurity from "/remote/sec_file"
```

7 online database コマンドを実行します。このコマンドは、必要に応じて sysaudits と sysauditoptions をアップグレードします。

```
online database sybsecurity
```

8 プラットフォームの『Adaptive Server Enterprise 設定ガイド』に従って、監査システム・プロシージャをロードします。

❖ **複数の *sysaudits* テーブルを *sybsecurity* に作成するには、次の手順に従います。**

1 追加テーブルを配置するデバイスを初期化します。

```
disk init name = "auditdev2",
    physname = "/dev/dsk/c2d0s6",
    size = "10M"
```

2 手順 1 で初期化したデバイスに sybsecurity データベースを拡張します。

```
alter database sybsecurity on auditdev2 = "2M"
```

3 sp_addaudittable システム・プロシージャを実行して、手順 1 で初期化したデバイス上に次の sysaudits テーブルを作成します。

```
sp_addaudittable auditdev2
```

4 各 sysaudits テーブルに対して、1 ～ 3 の手順を繰り返します。

監査証跡の管理の設定

監査証跡を効率的に管理するには、次の手順に従います。

1 監査機能が、個別のデバイスに配置された複数のテーブルを使用するようにインストールされていることを確認します。そうでない場合は、監査テーブルとデバイスを追加する必要があります。

2 スレッショルド・プロシージャを作成して、各監査テーブル・セグメントに付加します。

3 監査キュー・サイズと、現在の監査テーブルが満杯になった場合の適切な操作を示す設定パラメータを設定します。

以下の各項では、個別のデバイスに配置された複数テーブルを使用するように監査機能をインストールしたものと想定しています。監査テーブル用のデバイスが 1 つしかない場合は、「単一テーブル監査」(276 ページ) へ進んでください。

スレッショルド・プロシージャの設定

監査を有効にする前に、スレッショルド・プロシージャを設定して、現在のテーブルが満杯になったら監査テーブルを自動的に切り替えるようにしてください。

監査デバイス・セグメントのスレッショルド・プロシージャは、次のタスクを実行する必要があります。

- sp_configure を使用して current audit table 設定パラメータを設定し、次の空白の監査テーブルを現在のテーブルにする。
- insert...select コマンドを使用して、満杯に近づいた監査テーブルをアーカイブする。

現在の監査テーブルの変更

current audit table 設定パラメータは、Adaptive Server が監査ローを書き込むテーブルを設定します。システム・セキュリティ担当者は、sp_configure を実行して現在の監査テーブルを変更できます。構文は次のとおりです。*n* は、新しい現在の監査テーブルを指定する整数です。

```
sp_configure "current audit table", n
[, "with truncate"]
```

n の有効な値は次のとおりです。

- 1 は sysaudits_01、2 は sysaudits_02 を示します。
- 0 は、次のテーブルを自動的に現在の監査テーブルとして設定するように Adaptive Server に指示します。たとえば、インストール環境に 3 つの監査テーブル sysaudits_01、sysaudits_02、sysaudits_03 がある場合、現在の監査テーブルは次のように設定されます。
  - 現在の監査テーブルが sysaudits_01 の場合は 2
  - 現在の監査テーブルが sysaudits_02 の場合は 3
  - 現在の監査テーブルが sysaudits_03 の場合は 1

with truncate オプションは、新しいテーブルが空でない場合に、そのテーブルをトランケートすることを指定します。このオプションを指定しないと、テーブルが空になっていない場合、sp_configure コマンドは失敗します。

注意　Adaptive Server が現在の監査テーブルをトランケートしたときに、データがアーカイブ済みでなければ、そのテーブルの監査レコードは失われます。with truncate オプションを使用する前に、必ず監査データをアーカイブするようにしてください。

---

sp_configure を実行して現在の監査テーブルを変更するには、sso_role をアクティブにしてください。スレッショルド・プロシージャを作成して、現在の監査テーブルを自動的に変更することもできます。

監査テーブルのアーカイブ

select とともに insert を使用すると、sybsecurity 内の監査テーブルと同じカラムを持つ既存のテーブルに、監査データをコピーすることができます。

スレッショルド・プロシージャが、別のデータベース内のアーカイブ・テーブルにデータを正常にコピーできるようにするには、次の準備手順を実行してください。

1　sybsecurity 内の監査テーブルが存在するデバイスとは別のデバイス上に、アーカイブ・データベースを作成します。

2　sybsecurity の監査テーブルと同じカラムを持つアーカイブ・テーブルを作成します。このようなテーブルが存在しない場合は、select into を使用して where 句に false の条件を指定することによって、空のテーブルを作成することができます。次に例を示します。

```
use aud_db
go
select *
    into audit_data
    from sybsecurity.dbo.sysaudits_01
    where 1 = 2
```

where 条件は常に false です。したがって、sysaudits_01 の複製である空のテーブルが作成されます。

select into を使用する前に、アーカイブ・データベースで sp_dboption を使用して select into/bulk copy データベース・オプションをオンにしておく必要があります。

スレッショルド・プロシージャでは、sp_configure を使用して監査テーブルを変更した後で、insert と select を使用して、アーカイブ・データベース内のアーカイブ・テーブルにデータをコピーします。このプロシージャで実行するコマンドの例を示します。

```
insert aud_db.sso_user.audit_data
select * from sybsecurity.dbo.sysaudits_01
```

監査セグメント用スレッショルド・プロシージャの例

次のスレッショルド・プロシージャの例では、監査用に 3 つのテーブルが設定されているものと想定しています。

```
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)
```

各監査セグメントへのスレッショルド・プロシージャの付加

スレッショルド・プロシージャを各監査テーブル・セグメントに付加するには、システム・プロシージャ sp_addthreshold を使用します。

sp_addthreshold を実行する前に、必ず次のことを行ってください。

- インストール環境に合わせて設定する監査テーブルの数と、そのデバイス・セグメントの名前を決定する。
- sp_addthreshold の実行に必要な、スレッショルド・プロシージャに含まれるすべてのコマンドに対するパーミッションおよび役割を用意する。

警告！ sp_addthreshold と sp_modifythreshold は、sa_role を直接付与されたユーザだけがスレッショルドを追加または変更できるようにするために検査を行います。スレッショルドを追加または変更するときにアクティブなすべてのシステム定義の役割が、そのログインに有効な役割として、systhresholds テーブルに挿入されます。ただし、スレッショルド・プロシージャの起動時には、直接付与された役割だけがアクティブになります。

監査テーブルとそのセグメント

監査機能をインストールするとき、auditinit によって各監査テーブルの名前とそのセグメントが表示されます。セグメント名は、sysaudits_01 では “aud_seg1”、sysaudits_02 では “aud_seg2” というようになります。sybsecurity を現在のデータベースとして sp_helpsegment を実行すると、sybsecurity データベース内のセグメントに関する情報を検索できます。インストール環境の監査テーブル数を検索する方法の 1 つとして、次の SQL コマンドがあります。

```
use sybsecurity
go
select count(*) from sysobjects
    where name like "sysaudit%"
go
```

次の SQL コマンドを実行して、監査テーブルと sybsecurity データベースに関する詳細情報を取得することもできます。

```
sp_helpdb sybsecurity
go
use sybsecurity
go
sp_help sysaudits_01
go
sp_help sysaudits_02
go
  ...
```

必要な役割とパーミッション

sp_addthreshold は、データベース所有者かシステム管理者でなければ実行できません。通常は、システム・セキュリティ担当者が、sybsecurity データベースの所有者です。したがって、システム・セキュリティ担当者は sp_addthreshold を実行できます。また、sp_addthreshold を実行する権限に加えて、スレッショルド・プロシージャ内のすべてのコマンドの実行パーミッションが必要です。たとえば、sp_configure を実行して current audit table を設定するには、sso_role がアクティブでなければなりません。スレッショルド・プロシージャが起動すると、Adaptive Server は、sp_addthreshold の実行時に有効であったすべての役割とパーミッションをオンにしようとします。

3 つのデバイス・セグメントにスレッショルド・プロシージャ audit_thresh を付加する方法は次のとおりです。

```
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
```

このサンプル・スレッショルド・プロシージャ audit_thresh は、現在の監査テーブル内に残っている空きページが 250 よりも少なくなると、制御を受け取ります。

スレッショルド・プロシージャの追加の詳細については、『システム管理ガイド 第 2 巻』の「第 16 章 スレッショルドによる空き領域の管理」を参照してください。

サンプル・スレッショルド・プロシージャによる監査

監査が有効化されると、Adaptive Server は、すべての監査データを最初の現在の監査テーブルである sysaudits_01 に書き込みます。sysaudits_01 の空きページが 250 ページ以内になると、スレッショルド・プロシージャ audit_thresh が起動します。このプロシージャが現在の監査テーブルを sysaudits_02 に切り替えると、その直後から新しい監査レコードは sysaudits_02 に書き込まれます。また、このプロシージャは、sysaudits_01 のすべての監査データを audit_db にあるアーカイブ・テーブル audit_data にコピーします。監査テーブルの巡回は、このように手動介入なしで続きます。

設定パラメータの設定

監査機能をインストールした場合は、次の設定パラメータを設定してください。

- audit queue size は、メモリ内の監査キューのレコード数を設定します。
- suspend audit when device full は、現在の監査テーブルの空きがまったくなくなったときの Adaptive Server の動作を決定します。満杯状態は、現在のテーブル・セグメントに付加されたスレッショルド・プロシージャが正しく機能していない場合にのみ起こります。

監査キューの設定

監査キューのデフォルト・サイズは 100 バイトです。監査キュー・プールが使用するメモリ量は、audit queue size パラメータで定義され、メモリ・プールのデータ・バッファとオーバーヘッドが含まれます。ただし、プールのメモリ量はリリースとチップ・アーキテクチャ間で異なる場合があります。

監査キューの長さを設定するには、sp_configure を使用します。構文は次のとおりです。

```
sp_configure "audit queue size", [value]
```

value は、監査キューが保持できるレコードの数です。最小値は 1、最大値は 65,535 です。たとえば、次のコマンドは、監査キューのサイズを 300 に設定します。

```
sp_configure "audit queue size", 300
```

監査キュー・サイズとその他の設定パラメータを設定する方法については、『システム管理ガイド 第 1 巻』の「第 5 章 設定パラメータ」を参照してください。

デバイスが満杯の場合の監査の中断

複数の監査テーブルがそれぞれマスタ・デバイス以外の独立したデバイス上にあり、各監査テーブル・セグメントにスレッショルド・プロシージャが付加されていれば、監査デバイスが満杯になる状態は決して発生しません。スレッショルド・プロシージャが正常に機能していない場合だけ、「満杯」状態が発生します。デバイスが満杯になったときの処置を指定するには、sp_configure を使用して suspend audit when device full パラメータを設定します。次のいずれかのオプションを選択してください。

- 監査プロセスと、監査可能イベントを生成するすべてのユーザ・プロセスを中断します。システム・セキュリティ担当者が現在の監査テーブルをクリアしてから、通常の操作を再開します。
- 次の監査テーブルをトランケートし、そのテーブルの使用を開始します。これによって、システム・セキュリティ担当者の介入なしに通常の操作を進めることができます。

sp_configure to を使用して、この設定パラメータを設定します。また、sso_role をアクティブにする必要があります。構文は次のとおりです。

```
sp_configure "suspend audit when device full",
    [0|1]
```

- 0 を指定すると、現在の監査テーブルが満杯になったときは、次の監査テーブルがトランケートされ、そのテーブルが現在の監査テーブルとして使用されます。このパラメータを 0 に設定しても監査プロセスが中断することはありません。ただし、古い監査レコードは、アーカイブされていなければ完全に消失します。

- 1 (デフォルト値) を指定すると、監査プロセスと監査可能なイベントを生成するすべてのユーザ・プロセスが中断します。通常の操作を再開するには、システム・セキュリティ担当者がログインして、空のテーブルを現在の監査テーブルとして設定する必要があります。この間、システム・セキュリティ担当者は、通常の監査の対象外となります。通常の操作であれば監査レコードが生成されるようなアクションをシステム・セキュリティ担当者が実行すると、そのイベントに関するエラー・メッセージと情報が Adaptive Server のエラー・ログに送信されます。

スレッショルド・プロシージャが監査テーブル・セグメントに付加されている場合は、suspend audit when device full を 1 (on) に設定します。このパラメータを 0 (off) に設定すると、スレッショルド・プロシージャによって監査レコードがアーカイブされる前に、満杯の監査テーブルがトランケートされることがあります。

トランザクション・ログの管理の準備

この項では、sybsecurity 内のトランザクション・ログを管理するためのガイドラインを説明します。

trunc log on chkpt データベース・オプションがアクティブの場合は、自動 checkpoint の実行のたびに syslogs がトランケートされます。監査がインストールされると trunc log on chkpt の値は on になりますが、sp_dboption を使用すると、この値を変更できます。

トランザクション・ログのトランケーション

sybsecurity データベースに対して trunc log on chkpt オプションを有効にすれば、トランザクション・ログが満杯になることはありません。Adaptive Server がチェックポイントを実行するたびに、ログがトランケートされます。このオプションが有効の場合、dump transaction を使用してトランザクション・ログをダンプすることはできませんが、dump database を使用してデータベースをダンプできます。

「スレッショルド・プロシージャの設定」(268 ページ) の手順に従った場合は、監査テーブルは別のデータベース内のテーブルに自動的にアーカイブされます。このアーカイブ・データベースには、標準のバックアップとリカバリの手順を使用できます。

sybsecurity デバイスがクラッシュした場合は、データベースを再ロードして、監査を再開します。最悪の場合でも、メモリ内の監査キューと現在の監査テーブルが失われるだけで済みます。これは、アーカイブ・データベースにそれ以外の監査データがすべて含まれるためです。データベースを再ロードしたら、sp_configure with truncate を使用して、現在の監査テーブルを設定してトランケートします。

データベースをダンプした後に、サーバ全体の監査オプションを変更していなければ、sysauditoptions に保管されているすべての監査オプションが、sybsecurity の再ロード時に自動的にリストアされます。変更した場合は、監査を再開する前にスクリプトを実行してオプションを設定します。

トランケーションを使用しないトランザクション・ログの管理

db_option を使用して trunc log on chkpt をオフにすると、トランザクション・ログが満杯になる可能性があります。「ラストチャンス・スレッショルド・プロシージャ」をトランザクション・ログ・セグメントに付加することを検討してください。このプロシージャは、セグメントの空き領域が、Adaptive Server によって自動的に計算されるスレッショルドの量を下回ると起動されます。スレッショルド量は、トランザクション・ログのバックアップに必要な空きログ・ページ数から計算されます。

ラストチャンス・スレッショルド・プロシージャのデフォルト名は sp_thresholdaction ですが、sa_role がアクティブになっていれば sp_modifythreshold を使用して別の名前を指定できます。

注意　sp_modifythreshold は、“sa_role” がアクティブであることをチェックします。詳細については、「各監査セグメントへのスレッショルド・プロシージャの付加」(271 ページ) を参照してください。

Adaptive Server のデフォルトのプロシージャはありませんが、『システム管理ガイド 第 2 巻』の「第 16 章 スレッショルドによる空き領域の管理」にラストチャンス・スレッショルド・プロシージャの例が記載されています。このプロシージャは、dump transaction コマンドを実行して、ログをトランケートします。トランザクション・ログがラストチャンス・スレッショルド・ポイントに達すると、実行中のトランザクションは、領域が使用可能になるまで中断されます。トランザクションが中断されるのは、sybsecurity データベースに対して abort xact when log is full オプションが常に FALSE に設定されているためです。このオプションは変更できません。

trunc log on chkpt オプションを無効にすると、標準の手順で sybsecurity データベースのバックアップとリカバリを実行できますが、リストアされたデータベース内の監査テーブルが、デバイス障害発生時の状況と同期しない場合があることに注意してください。

監査の有効化と無効化

監査を有効または無効にするには、auditing 設定パラメータとともに sp_configure を使用します。構文は次のとおりです。

```
sp_configure "auditing", [0 | 1 ]
```

- 1 は監査を有効にします。
- 0 は監査を無効にします。

たとえば、監査を有効にするには、次のように入力します。

```
sp_configure "auditing", 1
```

注意 監査を有効にしたとき、または無効にしたときに、監査レコードが自動的に生成されます。表 8-5 (294 ページ) のイベント・コード 73 と 74 を参照してください。

単一テーブル監査

Sybase では、運用システムについては単一デバイスでの監査を使用しないよう強くおすすめします。使用する監査テーブルが 1 つだけの場合は、監査データをアーカイブしている間や監査テーブルをトランケートしている間に受信した監査レコードは失われるからです。単一の監査テーブルを使用している場合、これを防ぐ方法はありません。

使用する監査テーブルが 1 つだけの場合は、監査テーブルが満杯になる可能性が高くなります。監査テーブルが満杯になった場合の処置は、設定パラメータ suspend audit when device full の設定によって決まります。suspend audit when device full を on に設定すると、監査プロセスが中断し、監査可能イベントを生成するすべてのユーザ・プロセスも中断します。suspend audit when device full を off に設定すると、監査テーブルはトランケートされ、その監査テーブル内にあったすべての監査レコードが失われます。

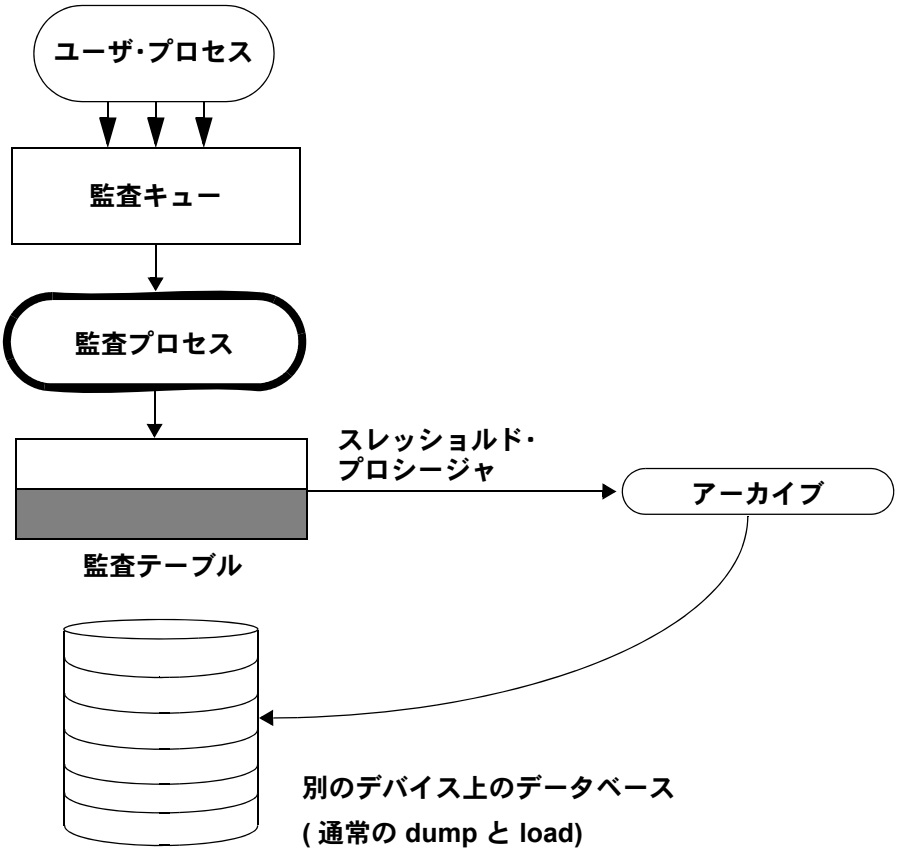
非運用システムの場合、少数の監査レコードの消失であれば、それほど問題はありません。このため、複数の監査テーブルを使用するためのディスク領域を追加できない場合や、使用できる追加デバイスがない場合は、単一テーブルを使用して監査を行います。

単一の監査テーブルを使用するための手順は、複数の監査テーブルを使用する場合に似ていますが、次の例外があります。

- インストール中、監査に使用するシステム・テーブルを 1 つだけ指定する。
- インストール中、監査システム・テーブル用のデバイスを 1 つだけ指定する。
- 監査レコードをアーカイブするために作成するスレッショルド・プロシージャは、複数の監査テーブルを使用する場合のものとは異なる。

図 8-2 は、監査プロセスが単一の監査テーブルを処理する方法を示します。

図 8-2: 単一の監査テーブルによる監査



単一テーブル監査の設定と管理

単一の監査テーブルを使用するための手順は、複数の監査テーブルを使用する場合と同じです。

単一テーブル監査の監査処理

単一テーブルでの監査の場合は、スレッショルド・プロシージャは次のタスクを実行する必要があります。

- insert コマンドと select コマンドを使用して、満杯に近づいた監査テーブルを別のテーブルにアーカイブする。
- truncate table コマンドを使用して、監査テーブルをトランケートし、新しい監査レコード用の領域を作成する。

監査レコードをアーカイブする前に、監査テーブルと同じカラムを持つアーカイブ・テーブルを作成します。この処理を終えると、スレッショルド・プロシージャで insert と select を使用して監査レコードをアーカイブ・テーブルにコピーすることができます。

次に、単一監査テーブルに使用するサンプル・スレッショルド・プロシージャを示します。

```
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
```

スレッショルド・プロシージャを作成したら、そのプロシージャを監査テーブル・セグメントに付加する必要があります。詳細については、「各監査セグメントへのスレッショルド・プロシージャの付加」(271 ページ) を参照してください。

警告！ マルチプロセッサ上では、監査テーブルが満杯になる前にトリガされるスレッショルド・プロシージャがあっても、監査テーブルが満杯になる可能性があります。たとえば、スレッショルド・プロシージャが負荷の重い CPU 上で実行されていて、監査可能なイベントを実行するユーザ・プロセスが負荷の比較的軽い CPU 上で実行されている場合、スレッショルド・プロシージャがトリガする前に、監査テーブルが満杯になる場合があります。設定パラメータ suspend audit when device full は、監査テーブルが満杯になったときの動作を指定します。このパラメータの設定方法については、「デバイスが満杯の場合の監査の中断」(273 ページ) を参照してください。

現在の監査テーブルが満杯になったときに起こる動作

現在の監査テーブルが満杯になると、次の動作が起こります。

1　監査プロセスは、テーブルに次の監査レコードを挿入しようとします。挿入はできないので、監査プロセスは終了します。エラー・メッセージは、エラー・ログに書き込まれます。

2　監査可能イベントをユーザが実行しようとしても、監査を進められないので、そのイベントは終了できません。ユーザ・プロセスは終了します。監査可能イベントを実行しないユーザは、影響を受けません。

3　ログイン監査が有効な場合は、システム・セキュリティ担当者以外は誰もサーバにログインできません。

4　sso_role をアクティブにして実行されるコマンドが監査対象の場合は、システム・セキュリティ担当者はコマンドを実行できません。

現在の監査テーブルが満杯になったときのリカバリの方法

現在の監査デバイスが満杯になって、監査キューも満杯である場合、システム・セキュリティ担当者の操作は監査の対象外となります。この時点から、システム・セキュリティ担当者によって監査可能イベントが実行されると、警告メッセージがエラー・ログ・ファイルに送信されます。このメッセージの内容は、日付と時刻、および監査が行われていないことを知らせる警告で、さらに、ログイン名、event コードなど、通常であれば監査テーブルの extrainfo カラムに保管される情報も含まれます。

現在の監査テーブルが満杯のとき、システム・セキュリティ担当者は、「監査テーブルのアーカイブ」(269 ページ) の説明に従って監査テーブルをアーカイブし、トランケートします。システム管理者が shutdown を実行してサーバを停止してから再起動すると、監査が再開します。

監査システムが異常終了した場合、システム・セキュリティ担当者は、現在の監査テーブルがアーカイブされてトランケートされた後でサーバを停止することができます。通常は、システム管理者だけが shutdown を実行できます。

監査の再起動

エラーが発生したために監査プロセスが強制的に終了された場合は、次のように入力することで sp_audit を手動で再起動できます。

```
sp_audit restart
```

監査プロセスの再起動は、現在実行中の監査がないことを条件として行うことができます。ただし、sp_configure “auditing” 1 の入力によって監査プロセスを有効にする必要があります。

グローバル監査オプションの設定

監査機能をインストールした後は、sp_audit を使用して監査オプションを設定できます。sp_audit の構文は次のとおりです。

sp_audit *option*, *login_name*, *object_name* [,*setting*]

sp_audit にパラメータを付けずに実行すると、すべてのオプションのリストが表示されます。sp_audit の詳細については、『リファレンス・マニュアル：プロシージャ』を参照してください。

注意 サーバの監査がアクティブ化されていないときは、監査は行われません。監査を起動する方法については、「監査の有効化と無効化」(276 ページ) を参照してください。

監査オプション：タイプと要件

sp_audit で *login_name* と *object_name* の各パラメータに指定できる値は、指定する監査オプションのタイプによって異なります。

- グローバル・オプションは、サーバのブートやディスク・コマンド、独自のユーザ定義監査レコードを可能にするかどうかなど、サーバ全体に影響するコマンドに適用されます。グローバル・イベントのオプション設定は、sybsecurity..sysauditoptions システム・テーブルに保存されます。
- データベース固有のオプションはデータベースに適用されます。データベース固有のオプションの例としては、データベースの変更、データベースへのデータのバルク・コピー (bcp in)、データベース内のオブジェクトへのアクセス権の付与および取り消し、データベース内へのオブジェクト作成などがあります。データベース固有のイベントのオプション設定は、master..sysdatabases システム・テーブルに保存されます。
- オブジェクト固有のオプションは、特定のオブジェクトに適用されます。オブジェクト固有のオプションの例としては、特定のテーブルやビューのローの選択、挿入、更新あるいは削除、および特定のトリガやプロシージャの実行などがあります。オブジェクト固有イベントのオプション設定は、関連するデータベース内の sysobjects システム・テーブルに保存されます。
- ユーザ固有のオプションは、特定のユーザあるいはシステム標準の役割に適用されます。ユーザ固有のオプションの例には、テーブルやビューへの特定のユーザのアクセス権や、特定のシステム標準の役割 (sa_role など) がアクティブな状態で実行されるすべてのアクションなどがあります。個々のユーザに関するオプション設定は、master..syslogins に保存されます。システム標準の役割の設定は、master..sysauditoptions に保存されます。

- 役割固有のオプションは、特定のユーザ、グループ、またはシステム役割に適用され、詳細なセキュリティ関連の監査を提供します。“role” 監査オプションはすべての役割関連のコマンドを監査し、create、alter、drop の各監査オプションは、役割定義コマンドの監査に使用されます。また、grant と revoke は、サブジェクトに対する役割の付与を監査するためのものです。master データベースは、オブジェクト名パラメータを必要とする監査オプションに指定されています。

表 8-2 は次のことを示します。

- option の有効な値および各オプションのタイプ － グローバル、データベース固有、オブジェクト固有、あるいはユーザ固有
- 各オプションの、*login_name* パラメータおよび *object_name* パラメータの有効な値
- 監査オプションの設定時に使用するデータベース
- そのオプションを設定したときに監査を受けるコマンドまたはアクセス
- 各オプションの例

すべてのオプションは、デフォルトでオフになっています。

表 8-2: 監査オプション、要件および例

オプション (オプションのタイプ)	*login_name*	*object_name*	オプション設定時に使用するデータベース	監査されるコマンドまたはアクセス
adhoc (ユーザ固有)	all	all	任意	ユーザは sp_addauditrecord を使用できる
	次の例では独自のユーザ定義監査レコードを使用可能にします。 `sp_audit "adhoc", "all", "all", "on"`			
all (ユーザ固有)	ログイン名または役割	all	任意	特定ユーザによるすべてのアクション、または特定の役割をアクティブにしたユーザによるすべてのアクション
	次の例では sa_role がアクティブになっているすべてのアクションについて監査をオンにします。 `sp_audit "all", "sa_role", "all", "on"`			
alter (データベース固有)	all	監査されるデータベース	任意	alter database、alter role、alter table
	次の例では master データベース内の alter database および alter table のすべての実行について監査をオンにします。 `sp_audit @option = "alter", @login_name = "all", @object_name = "master", @setting = "on"`			
bcp (データベース固有)	all	監査されるデータベース	任意	bcp in
	次の例では pubs2 データベースでの bcp の監査のステータスを返します。 `sp_audit "bcp", "all", "pubs2"` *setting* の値が指定されていない場合は、指定されたオプションの監査のステータスを返す			

オプション (オプションのタイプ)	*login_name*	*object_name*	オプション設定時に使用するデータベース	監査されるコマンドまたはアクセス
bind (データベース固有)	all	監査されるデータベース	任意	sp_bindefault、sp_bindmsg、sp_bindrule
	次の例では planning データベースの bind 監査をオフにします。 `sp_audit "bind", "all", "planning", "off"`			
cmdtext (ユーザ固有)	監査されるユーザのログイン名	all	任意	ユーザによって入力された SQL テキスト (該当するテキストがパーミッション検査に合格したかどうかは反映されない。*eventmod* の値は常に 1)
	次の例ではデータベース所有者の text 監査をオフにします。 `sp_audit "cmdtext", "sa", "all", "off"`			
create (データベース固有)	all	監査されるデータベース	任意	create database、create table、create role、create procedure、create trigger、create rule、create default、sp_addmessage、create view、create index、create function
	注意 create database を監査する場合は、*object_name* にmaster を指定する。これにより master 内の他のオブジェクトの作成も監査対象となる。 次の例では planning データベース内で正常に行われたオブジェクト作成の監査をオンにします。 `sp_audit "create", "all", "planning", "pass"` master データベースを指定していないため、create database の監査の現在のステータスは影響を受けない			
dbaccess (データベース固有)	all	監査されるデータベース	任意	他のデータベースからこのデータベースへのすべてのアクセス
	次の例では project データベースへの外部からのアクセスをすべて監査します。 `sp_audit "dbaccess", "all", "project", "on"`			
dbcc テーブル (グローバル)	all	all	任意	パーミッションを必要とするすべての dbcc コマンド
	次の例では dbcc コマンドのすべての実行を監査します。 `sp_audit "dbcc", "all", "all", "on"`			

オプション (オプションのタイプ)	*login_name*	*object_name*	オプション設定時に使用するデータベース	監査されるコマンドまたはアクセス
delete (オブジェクト固有)	all	監査の対象となるテーブルまたはビューの名前、または default view か default table	テーブルまたはビューのデータベース (tempdb を除く)	テーブルからの delete、ビューからの delete
	次の例では現在のデータベース内の将来のテーブルすべてについて、すべての削除アクションを監査します。 `sp_audit "delete", "all", "default table", "on"`			
disk (グローバル)	all	all	任意	disk init、disk refit、disk reinit、disk mirror、disk unmirror、disk remirror、disk resize
	次の例ではサーバのすべてのディスク・アクションを監査します。 `sp_audit "disk", "all", "all", "on"`			
drop (データベース固有)	all	監査されるデータベース	任意	drop database、drop table、drop role、drop procedure、drop index、drop trigger、drop rule、drop default、sp_dropmessage、drop view、drop function
	次の例では financial データベース内の、パーミッション検査に不合格となったすべての drop コマンドを監査します。 `sp_audit "drop", "all", "financial", "fail"`			
dump (データベース固有)	all	監査されるデータベース	任意	dump database、dump transaction
	次の例では pubs2 データベース内のダンプ・コマンドを監査します。 `sp_audit "dump", "all", "pubs2", "on"`			
encryption_key (データベース固有)	all	監査されるデータベース	任意	alter encryption key create encryption key drop encryption key sp_encryption
	次の例では pubs2 データベースで指定した上記すべてのコマンドを監査します。 `sp_audit "encryption_key", "all", "pubs2", "on"`			
errors (グローバル)	all	all	任意	致命的なエラー、致命的ではないエラー
	次の例ではサーバからのエラーを監査します。 `sp_audit "errors", "all", "all", "on"`			
errorlog	all	all	任意	sp_errorlog 関数または errorlog_admin 関数
	次の例では「ログの変更」によって新しい Adaptive Server エラー・ログ・ファイルに移動する試みを監査します。 `sp_audit "errorlog", "all", "all", "on"`			

オプション (オプションのタイプ)	*login_name*	*object_name*	オプション設定時に使用するデータベース	監査されるコマンドまたはアクセス
exec_procedure (オブジェクト固有)	all	監査の対象となるプロシージャの名前または default procedure	プロシージャのデータベース (tempdb を除く)	execute
	次の例では現在のデータベース内の新しいプロシージャの自動監査をオフにします。 `sp_audit "exec_procedure", "all", "default procedure", "off"`			
exec_trigger (オブジェクト固有)	all	監査の対象となるトリガの名前または default trigger	トリガのデータベース (tempdb を除く)	トリガを起動するすべてのコマンド
	次の例では現在のデータベース内のトリガ trig_fix_plan の失敗した実行をすべて監査します。 `sp_audit "exec_trigger", "all", "trig_fix_plan", "fail"`			
func_dbaccess (データベース固有)	all	監査の対象となるデータベースの名前	任意	次の関数を使用したデータベースへのアクセス：curunreserved_pgs、db_name、db_id、lct_admin、setdbrepstat、setrepstatus、setrepdefmode、is_repagent_enabled、rep_agent_config、rep_agent_admin
	次の例では組み込み関数による strategy データベースへのアクセスを監査します。 `sp_audit @option="func_dbaccess", @login_name="all", @object_name = "strategy", @setting = "on"`			
func_obj_access (オブジェクト固有)	all	sysobjects にエントリがあるオブジェクトの名前	任意	次の関数を使用したオブジェクトへのアクセス：schema_inc、col_length、col_name、data_pgs、index_col、object_id、object_name、reserved_pgs、rowcnt、used_pgs、has_subquery
	次の例では組み込み関数による customer テーブルへのアクセスを監査します。 `sp_audit @option="func_obj_access", @login_name="all", @object_name = "customer", @setting = "on"`			
grant (データベース固有)	all	監査の対象となるデータベースの名前	任意	grant
	次の例では planning データベース内のすべての権限の付与を監査します。 `sp_audit @option="grant", @login_name="all", @object_name = "planning", @setting = "on"`			

オプション (オプションのタイプ)	*login_name*	*object_name*	オプション設定時に使用するデータベース	監査されるコマンドまたはアクセス
insert (オブジェクト固有)	all	ローを挿入するビューまたはテーブルの名前、または default view か default table	オブジェクトのデータベース (tempdb を除く)	テーブルへの insert、ビューへの insert
	次の例では現在のデータベース内の dpt_101_view ビューへの、すべての挿入を監査します。 `sp_audit "insert", "all", "dpt_101_view", "on"`			
install (データベース固有)	all	監査されるデータベース	任意	install java
	次の例では planning データベースへの java クラスのインストールを監査します。 `sp_audit "install", "all", "planning", "on"`			
load (データベース固有)	all	監査されるデータベース	任意	load database、load transaction
	次の例では projects_db データベース内の、失敗したすべてのデータベース・ロードおよびトランザクション・ロードの実行を監査します。 `sp_audit "load", "all", "projects_db", "fail"`			
login (グローバル)	all	all	任意	Adaptive Server へのログイン
	次の例ではサーバへの失敗したログイン試行をすべて監査します。 `sp_audit "login", "all", "all", "fail"`			
login_locked (グローバル)	all	all	任意	
	次の例では設定されているログイン試行失敗回数を超えているため、ログインがロックされることを示します。 `sp_audit "login_locked", "all", "all", "on"`			
logout	all	all	任意	Adaptive Server からのログアウト
	次の例では、サーバからのすべてのログアウトについて監査をオフにします。 `sp_audit "logout", "all", "all", "off"`			
mount (グローバル)	all	all	任意	mount database
	次の例では、発行されたすべての mount database コマンドを監査します。 `sp_audit "mount", "all", "all", "on"`			
password	all	all	任意	グローバル・パスワードおよびログイン・ポリシー・オプションの設定
	次の例では、パスワードの監査をオンにします。 `sp_audit "password", "all", "all", "on"`			
quiesce (グローバル)	all	all	任意	quiesce database
	次の例では、quiesce database コマンドに対する監査をオンにします。 `sp_audit "quiesce", "all", "all", "on"`			

オプション (オプションのタイプ)	*login_name*	*object_name*	オプション設定時に使用するデータベース	監査されるコマンドまたはアクセス
reference (オブジェクト固有)	all	ローを挿入するビューまたはテーブルの名前、または default view か default table	任意	create table、alter table
	次の例では、titles テーブルへの参照の作成に対する監査をオフにします。 `sp_audit "reference", "all", "titles", "off"`			
remove (データベース固有)	all	all	任意	Java クラスの削除を監査する。
	次の例では planning データベースでの Java クラスの削除を監査します。 `sp_audit "remove", "all", "planning", "on"`			
revoke (データベース固有)	all	監査されるデータベース	任意	revoke
	次の例では、payments_db データベース内の revoke の実行の監査をオフにします。 `sp_audit "revoke", "all", "payments_db", "off"`			
rpc (グローバル)	all	all	任意	リモート・プロシージャ・コール (受信と発信の両方)
	次の例では、サーバから、あるいはサーバへのすべてのリモート・プロシージャ・コールを監査します。 `sp_audit "rpc", "all", "all", "on"`			
security (グローバル)	all	all	任意	サーバ全体のすべてのセキュリティ関連イベント。表 8-5 の “security” オプションを参照。
	次の例では、サーバ全体でのセキュリティ関連イベントを監査します。 `sp_audit "security", "all", "all", "on"`			
select (オブジェクト固有)	all	ローを挿入するビューまたはテーブルの名前、または default view か default table	オブジェクトのデータベース (tempdb を除く)	テーブルからの select、ビューからの select
	次の例では、現在のデータベース内の customer テーブルからの、失敗したすべての選択を監査します。 `sp_audit "select", "all", "customer", "fail"`			
setuser (データベース固有)	all	all	任意	setuser
	次の例では、projdb データベース内の setuser の実行をすべて監査します。 `sp_audit "setuser", "all", "projdb", "on"`			

オプション (オプションのタイプ)	*login_name*	*object_name*	オプション設定時に使用するデータベース	監査されるコマンドまたはアクセス
table_access (ユーザ固有)	監査されるユーザのログイン名。	all	任意	テーブル内での select、delete、update または insert によるアクセス
	次の例では、“smithson” というログインによるすべてのテーブル・アクセスを監査します。 `sp_audit "table_access", "smithson", "all", "on"`			
transfer_table (グローバル)	all	all	任意	サーバワイドなオプション。sysauditoptions には表示されない。
	次の例では、サーバ全体での転送関連イベントを監査します。 `sp_audit "transfer_table", "tdb1.table1", "all", "on"`			
truncate (データベース固有)	all	監査されるデータベース	任意	truncate table
	次の例では、customer データベース内のすべてのテーブル・トランケーションを監査します。 `sp_audit "truncate", "all", "customer", "on"`			
unbind (データベース固有)	all	監査されるデータベース	任意	sp_unbindefault、sp_unbindrule、sp_unbindmsg
	次の例では、master データベース内の、失敗したすべてのバインド解除試行を監査します。 `sp_audit "unbind", "all", "master", "fail"`			
unmount (グローバル)	all	all	任意	unmount database
	次の例では、任意のデータベースでマニフェスト・ファイルをマウント解除または作成しようとするすべての試みを監査します。 `sp_audit "unmount", "all", "all", "on"`			
update (オブジェクト固有)	all	監査の対象となるオブジェクトの名前、または default table か default view	オブジェクトのデータベース (tempdb を除く)	テーブルへの update、ビューへの update
	次の例では、ユーザによる、現在のデータベース内の projects テーブル更新の試行をすべて監査します。 `sp_audit "update", "all", "projects", "on"`			
view_access (ユーザ固有)	監査されるユーザのログイン名	all	任意	ビューへの select、delete、insert または update
	次の例では、“joe” というユーザのビュー監査をオフにします。 `sp_audit "view_access", "joe", "all", "off"`			

監査オプションの設定の例

company_operations データベース内の projects テーブル、およびそのデータベース内の新しいテーブルすべてに対して失敗したすべての delete を監査する場合を想定します。projects テーブルの監査にはオブジェクト固有の delete オプションを使用し、データベース内の今後作成されるすべてのテーブルの監査には default table オプションを使用します。オブジェクト固有の監査オプションを設定するには、sp_audit を実行する前にそのオブジェクトのデータベースに移動する必要があります。

```
sp_audit "security", "all", "all", "fail"
```

この例では、次のコマンドを実行します。

```
use company_operations
go
sp_audit "delete", "all", "projects", "fail"
go
sp_audit "delete", "all", "default table",
"fail"
go
```

役割定義の監査

例 1 次のように、役割変更の監査をオンにします。

```
sp_audit "alter", "all", "master", "pass"
```

例 2 次のように、正常な役割作成の監査をオンにします。

```
sp_audit "alter", "all", "master", "on"
```

例 3 次の例では、役割削除の監査をオフにします。

```
sp_audit "drop", "all", "master", "off"
```

例 4 次のように、役割付与の監査をオフにします。

```
sp_audit "grant", "all", "master", "off"
```

監査は、AUD_EVT_UDR_CMD (85) イベント監査レコードを生成する、grant または role 監査オプションを使用して実行されます。

例 5 次のように、役割取り消しの監査をオンにします。

```
sp_audit "revoke", "all", "master", "on"
```

監査は、AUD_EVT_UDR_CMD (85) イベント監査レコードを生成する、revoke または role 監査オプションを使用して実行されます。

システム・ストアド・プロシージャとコマンドのパスワード・パラメータを隠す

監査が設定されて有効になっているとき、sp_audit にオプションの 'cmdtext' が設定されていると、監査ログ内の監査レコードではシステム・ストアド・プロシージャとコマンドのパスワード・パラメータが固定長のアスタリスク文字列で置き換えられます。

たとえば、監査が有効になっていて sp_audit cmdtext が設定されている場合の出力は次のようになります。

```
alter login johnd with password oldpasswd modify password
'newpasswd'
```

コマンドから次のような出力が返されます。

```
alter login johnd with password ****** modify password '******'
```

これで監査ログにアクセスできる他のユーザにパスワードを見られる心配がなくなります。

現在の監査設定の判別

指定オプションに関する現在の監査設定を判別するには、sp_displayaudit を使用します。構文は次のとおりです。

```
sp_displayaudit [procedure | object | login | database | global |
default_object | default_procedure [, name]]
```

詳細については、『リファレンス・マニュアル：プロシージャ』の「sp_displayaudit」を参照してください。

監査証跡へのユーザ指定レコードの追加

sp_addauditrecord を使用すると、ユーザは、監査証跡にコメントを入力できます。構文は次のとおりです。

```
sp_addauditrecord [text] [, db_name] [, obj_name]
    [, owner_name] [, dbid] [, objid]
```

パラメータはすべて省略可能です。

- *text* は、監査テーブル extrainfo に追加するメッセージのテキストです。
- *db_name* は、レコードで参照されるデータベースの名前です。これは、現在の監査テーブルの dbname カラムに挿入されます。
- *obj_name* は、レコードで参照されるオブジェクトの名前です。これは、現在の監査テーブルの objname カラムに挿入されます。
- *owner_name* は、レコードで参照されるオブジェクトの所有者です。これは、現在の監査テーブルの objowner カラムに挿入されます。

- *dbid* は、db_name のデータベース ID 番号を表す整数値です。これは、現在の監査テーブルの dbid カラムに挿入されます。引用符で囲まないでください。
- *objid* は、obj_name のオブジェクト ID 番号を表す整数値です。引用符で囲まないでください。*objid* は、現在の監査テーブルの objid カラムに挿入されます。

sp_addauditrecord は次の場合に使用できます。

- 実行するユーザが、sp_addauditrecord に対する実行パーミッションを持っている。
- 監査設定パラメータが sp_configure によってアクティブ化されている。
- adhoc 監査オプションが sp_audit によって有効化されている。

デフォルトでは、システム・セキュリティ担当者と sybsecurity のデータベース所有者のみが sp_addauditrecord を使用できます。その実行パーミッションは別のユーザに付与できます。

ユーザ定義監査レコードの追加例

次の例では、現在の監査テーブルにレコードを追加します。テキスト部分は現在の監査テーブルの extrainfo カラムに、“corporate” は dbname カラムに、“payroll” は objname カラムに、“dbo” は objowner カラムに、“10” は dbid カラムに、“1004738270” は objid カラムにそれぞれ挿入されます。

```
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
```

次の例は、現在の監査テーブルの extrainfo カラムと dbname カラムにだけ情報を挿入します。

```
sp_addauditrecord @text="I am disabling auditing
briefly while we reconfigure the system",
@db_name="corporate"
```

監査証跡のクエリ

監査証跡を問い合わせるには、SQL を使用して、監査データを選択および要約します。「監査証跡の管理の設定」(268 ページ) で説明している手順に従った場合は、監査データは別のデータベース内の 1 つまたは複数のテーブルに自動的にアーカイブされます。たとえば、監査データが、audit_db データベースの audit_data というテーブル内にあるとします。この場合、“bob” によって 1993 年 7 月 5 日に実行されたタスクの監査レコードを選択するには、次のコマンドを実行します。

```
use audit_db
go
select * from audit_data
   where loginname = "bob"
   and eventtime like "Jul 5% 93"
go
```

次のコマンドでは、システム・セキュリティ担当者の役割がアクティブなユーザによって、pubs2 データベースで実行されたコマンドの監査レコードを要求します。

```
select * from audit_data
   where extrainfo like "%sso_role%"
   and dbname = "pubs2"
go
```

次のコマンドでは、すべてのテーブル・トランケーション (イベント 64) の監査レコードを要求します。

```
select * from audit_data
   where event = 64
go
```

監査イベントの名前を使用して監査証跡を問い合わせるには、audit_event_name 関数を使用します。たとえば、すべてのデータベース作成イベントに対する監査レコードを要求するには、次のように入力します。

```
select * from audit_data where audit_event_name(event)
   = "Create Database"
go
```

監査テーブルの概要

システム監査テーブルにアクセスできるのはシステム・セキュリティ担当者だけで、システム・セキュリティ担当者は SQL コマンドを実行してテーブルを読み込むことができます。システム監査テーブルに対して使用できるコマンドは、select と truncate だけです。

表 8-3 は、すべての監査テーブルにあるカラムの説明です。

表 8-3: 各監査テーブル内のカラム

カラム名	データ型	説明
event	smallint	監査されるイベントのタイプ。表 8-5 (294 ページ) を参照してください。
eventmod	smallint	監査されるイベントに関する詳細。該当するイベントがパーミッション検査に合格したかどうかを示す。値は次のとおり。 • 0 = このイベントの修飾子はない。 • 1 = イベントがパーミッションの検査に成功した。 • 2 = イベントがパーミッションの検査に失敗した。
spid	smallint	監査レコードの書き込みが発生したプロセスのサーバ・プロセス ID。
eventtime	datetime	監査イベントが起こった日付と時刻。
sequence	smallint	単一イベント内のレコードのシーケンス番号。一部のイベントは、複数の監査レコードを必要とする。
suid	smallint	監査イベントを実行したユーザのサーバ・ログイン ID。
dbid	int null	監査されるイベントが発生したデータベースの ID、または、オブジェクト、ストアド・プロシージャ、トリガが存在するデータベースの ID (イベントのタイプによる)。
objid	int null	アクセスされたオブジェクト、ストアド・プロシージャ、またはトリガの ID。
xactid	binary(6) null	監査イベントを含むトランザクション ID。マルチデータベース・トランザクションの場合は、トランザクションが開始したデータベースからのトランザクション ID。
loginname	varchar(30) null	suid に対応するログイン名。
dbname	varchar(30) null	dbid に対応するデータベース名。
objname	varchar(30) null	objid に対応するオブジェクト名。
objowner	varchar(30) null	objid の所有者名。
extrainfo	varchar(255) null	監査イベントについての追加情報。このカラムに格納される一連の項目は、セミコロンで区切られている。詳細については、「extrainfo カラムの読み込み」(293 ページ) を参照してください。
nodeid	tinyint	イベントが発生したクラスタ内のサーバの nodeid

extrainfo カラムの読み込み

extrainfo カラムには、一連のデータがセミコロンで区切られて格納されています。このデータは、次のカテゴリから構成されます。

表 8-4: extrainfo カラム内の情報

位置	カテゴリ	説明
1	役割	アクティブな役割をブランクで区切ったリスト。
2	キーワードまたはオプション	イベントに使用されたキーワードまたはオプションの名前。たとえば、alter table コマンドでは、add column オプションや drop constraint オプションなどが使用される。複数のキーワードまたはオプションの場合は、カンマで区切られる。
3	以前の値	イベントによって値が更新された場合は、更新される前の値がこの項目に格納される。
4	現在の値	イベントによって値が更新された場合は、新しい値がこの項目に格納される。
5	その他の情報	イベントについて記録された、セキュリティ関連のその他の情報。
6	代理権限情報	set proxy が有効なときにイベントが発生した場合は、元のログイン名が格納される。
7	プリンシパル名	ユーザのログインがセキュア・デフォルト・ログインであり、ユーザが統一化ログインを介して Adaptive Server にログインした場合に、基本となるセキュリティ・メカニズムのプリンシパル名が格納される。セキュア・デフォルト・ログインが使用されていない場合、この項目の値は NULL。

次の例は、監査設定パラメータを変更するイベントの extrainfo カラムを示します。

```
sso_role;suspend audit when device full;1;0;;ralph;
```

このエントリは、システム・セキュリティ担当者が、設定パラメータ suspend audit when device full を 1 から 0 に変更したことを示します。このエントリに “Other information” はありません。6 番目のカテゴリは、ユーザ “ralph” が代理ログインによって操作していたことを示します。プリンシパル名はありません。

監査レコードの他のフィールドには、他の関連情報が格納されます。たとえば、サーバ・ユーザ ID (suid) とログイン名 (loginname) もレコードに含まれています。

表 8-5 は、event カラムに表示される値を sp_audit のオプション順にリストにしたものです。「extrainfo 出力の情報」の欄では、監査テーブルの extrainfo カラムに表示される情報を、表 8-4 に示すカテゴリに基づいて説明しています。

表 8-5: event カラムと extrainfo カラムの値

監査オプション	監査されるコマンドまたはアクセス	イベント	extrainfo の情報
(オプションによって制御されるのではなく、自動的に監査されるイベント)	監査の有効化に使用するコマンド：sp_configure auditing	73	—
(オプションによって制御されるのではなく、自動的に監査されるイベント)	監査の無効化に使用するコマンド：sp_configure auditing	74	—
管理者のアカウントのロック解除	監査の無効化に使用するコマンド：sp_configure auditing	74	—
adhoc	ユーザ定義監査レコード	1	extrainfo は、sp_addauditrecord の text パラメータによって埋められる
alter	alter database	2	サブコマンド・キーワード： alter maxhold alter size inmemory
	alter...modify owner *name_in_db*	124	サブコマンド・キーワード： • ユーザ定義型の場合：*owner. obj_name* オプションが指定されている場合は *name_in_db* preserve permissions。 • オブジェクトの場合：オプションが指定されている場合は *name_in_db* preserve permission。
	alter...modify owner *login_name* as concrete_owner	124	サブコマンド・キーワード： ユーザ定義型には適用されない。 オブジェクトの場合： オプションが指定されている場合は *login_name* preserve permissions。
	alter table	3	サブコマンド・キーワード： add/drop/modify column replace columns replace decrypt default replace/add decrypt default add constraint drop constraint 1 つまたは複数の暗号化カラムが追加される場合、extrainfo には次が含まれる。ここで、*keyname* はキーの完全に修飾された名前。 add/drop/modify column *column1*/*keyname1*, [,*column2*/*keyname2*]
bcp	bcp in	4	—

監査オプション	監査されるコマンドまたはアクセス	イベント	extrainfo の情報
bind	sp_bindefault	6	その他の情報：デフォルトの名前
	sp_bindmsg	7	その他の情報：メッセージ ID
	sp_bindrule	8	その他の情報：ルールの名前
all、create	create database	9	キーワードまたはオプション：inmemory
cmdtext	すべてのコマンド	92	クライアントによって送信されるコマンドのテキスト
create	create database	9	—
	create default	14	—
	create procedure	11	—
	create rule	13	—
	create table	10	暗号化カラムでは、extrainfo にはカラム名とキー名が含まれます。 EK *column1*/*keyname1*[,*column2 keyname2*] このとき、EK は、後続の情報が暗号化キーを参照することを示すプレフィクスです。また、*keyname* はキーの完全修飾名です。
	create trigger	12	—
	create view	16	—
	create index	104	その他の情報：インデックスの名前
	create function	97	—
	sp_addmessage	15	その他の情報：メッセージ番号
dbaccess	すべてのユーザによるデータベースへのあらゆるアクセス	17	キーワードまたはオプション： use cmd outside reference
dbcc	dbcc すべてのキーワード	81	キーワードまたはオプション：checkstorage などの dbcc のキーワードとそのキーワードのオプション。
delete	テーブルからの delete	18	キーワードまたはオプション：delete
	ビューからの delete	19	キーワードまたはオプション：delete

監査オプション	監査されるコマンドまたはアクセス	イベント	extrainfo の情報
disk	disk init	20	キーワードまたはオプション：disk init その他の情報：ディスクの名前
	disk mirror	23	キーワードまたはオプション：disk mirror その他の情報：ディスクの名前
	disk refit	21	キーワードまたはオプション：disk refit その他の情報：ディスクの名前
	disk reinit	22	キーワードまたはオプション：disk reinit その他の情報：ディスクの名前
	disk release	87	キーワードまたはオプション：disk release その他の情報：ディスクの名前
	disk remirror	25	キーワードまたはオプション：disk remirror その他の情報：ディスクの名前
	disk unmirror	24	キーワードまたはオプション：disk unmirror その他の情報：ディスクの名前
	disk resize	100	キーワードまたはオプション：disk resize その他の情報：ディスクの名前
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	その他の情報：インデックス名
	drop function	98	—
	sp_dropmessage	32	その他の情報：メッセージ番号
dump	dump database	34	—
	dump transaction	35	—

encryption_key	sp_encryption	106	パスワードを初めて設定した場合：
			`ENCR_ADMIN system_encr_passwd password ********`
			パスワードを後日変更した場合：
			`ENCR_ADMIN system_encr_passwd password ******** ********`

	create encryption key	107	キーワードの内容は次のとおりです。 algorithm name-bitlength/IV [random|NULL]/pad [random |NULL] user/system 次に例を示します。 `AES-128/IV RANDOM/PAD NULL USER`

監査オプション	監査されるコマンドまたはアクセス	イベント	extrainfo の情報
encryption_key	alter encryption key	108	`default/not default`
	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 *keyvalue* は、alter encryption key modify encryption の複写についてのみ表示されます。たとえば、ユーザ “stephen” がそのキー・コピーを変更すると、次の情報が保存されます。 `MODIFY ENCRYPTION for user stephen WITH USER PASSWD`
	AEK add encryption	119	add encryption for user *user_name* for login association | recovery|with keyvalue] *keyvalue* は、alter encryption key add encryption の複写についてのみ表示されます。
	alter encryption key drop encryption	120	drop encryption [for recovery | for user *user_name* 『暗号化カラム・ユーザーズ・ガイド』を参照してください。
	alter encryption key modify owner	121	modify owner [new owner *user_name*] 『暗号化カラム・ユーザーズ・ガイド』を参照してください。
	alter encryption key recover key	122	recovery key [with *key_value*] with *keyvalue* は、alter encryption key の複写時にのみ使用されます。 『暗号化カラム・ユーザーズ・ガイド』を参照してください。
errorlog	errorlog 関数または errorlog_admin 関数	127	errorlog_admin に渡されたパラメータは、サブコマンドの特定のために記録されます：errorlog_admin (param1, param2,...)
errors	致命的なエラー	36	その他の情報：*Error number*.*Severity*.*State*
	致命的ではないエラー	37	その他の情報：*Error number*.*Severity*.*State*
exec_procedure	プロシージャの実行	38	その他の情報：すべての入力パラメータ
exec_trigger	トリガの実行	39	—
func_obj_access、func_dbaccess	Transact-SQL 関数を介したオブジェクトおよびデータベースへのアクセス (関数を監査するには、sa_role について監査を有効にする必要があります)。	86	—

監査オプション	監査されるコマンドまたはアクセス	イベント	extrainfo の情報
grant	grant	40	使用可能な場合は完全なコマンド・テキストを含む。そうでない場合は権限を付与するユーザとコマンド・タイプを含む。
insert	テーブルへの insert	41	キーワードまたはオプション： • insert を使用する場合：insert • select into を使用する場合：insert into の後ろに完全修飾されたオブジェクト名が続く
	ビューへの insert	42	キーワードまたはオプション：insert
install	install	93	—
load	load database	43	—
	load transaction	44	—
login	サーバへのログインすべて	45	その他の情報： • ログインが行われたマシンのホスト名と IP アドレス • 失敗したログインの *Error number.Severity.State*
login_locked	ログインの設定失敗回数を超えているため、ログインはロックされる。	112	
logout	サーバからのログアウトすべて	46	その他の情報：ホスト名
mount	mount database	101	—
password	sp_passwordpolicy と、list 以外のそのすべてのアクション。	115	sp_passwordpolicy のパラメータ
quiesce	quiesce database	96	—
reference	テーブルへの参照の作成	91	キーワードまたはオプション：reference その他の情報：参照するテーブルの名前
remove	remove java	94	—
revoke	revoke	47	使用可能な場合は完全なコマンド・テキストを含む。そうでない場合は権限を付与するユーザとコマンド・タイプを含む。
rpc	別のサーバからのリモート・プロシージャ・コール	48	キーワードまたはオプション：クライアント・プログラムの名前 その他の情報：サーバ名、すなわち RPC が実行されたマシンのホスト名
	別のサーバへのリモート・プロシージャ・コール	49	キーワードまたはオプション：プロシージャ名
role locked	役割の設定/設定解除	133	役割の名前とロックの理由 • alter role *rolename* lock を手動で実行することにより suid によって役割がロックされます。 • 役割アクティブ化の試行回数が max failed_logins に達したため Adaptive Server によって役割がロックされました。

監査オプション	監査されるコマンドまたはアクセス	イベント	extrainfo の情報
security	connect to (CIS のみ)	90	キーワードまたはオプション：connect to
	online database	83	—
	proc_role 関数 (システム・プロシージャ内での実行)	80	その他の情報：必要な役割
	SSO によるパスワードの再生成	76	キーワードまたはオプション：SSO パスワードの設定 その他の情報：ログイン名
	役割のオンとオフ	55	以前の値：on または off 現在の値：on または off その他の情報：設定された役割の名前
	サーバの起動	50	その他の情報： -d*masterdevicename* -i*interfaces file path* -S*servername* -e*errorfilename*
	sp_webservices	111	キーワードまたはオプション：単一の Web サービスを配備する場合は deploy、すべての Web サービスを配備する場合は deploy_all
	sp_webservices	111	キーワードまたはオプション：単一の Web サービスの配備を解除する場合は undeploy、すべての Web サービスの配備を解除する場合は undeploy_all
	サーバの停止	51	キーワードまたはオプション：shutdown
	set proxy または set session authorization	88	以前の値：以前の suid 現在の値：新しい suid
	sp_configure	82	キーワードまたはオプション：SETCONFIG その他の情報： • パラメータが設定される場合は、設定パラメータの数 • 設定ファイルを使用してパラメータを設定する場合は、設定ファイルの名前
	sp_ssladmin 管理の有効化	99	証明書を追加する場合は、SSL_ADMIN addcert を含むキーワード
	監査テーブルへのアクセス	61	—
	create login、drop login	103	キーワードまたはオプション：create login、drop login
	create、drop、alter、grant、revoke role	85	キーワードまたはオプション：create、drop、alter、grant、または revoke role
	組み込み関数	86	キーワードまたはオプション：関数の名前

監査オプション	監査されるコマンドまたはアクセス	イベント	extrainfo の情報
	監査の対象となるセキュリティ・コマンドまたはアクセス。特に、管理者のアカウントをロック解除するための -u オプションを使用した Adaptive Server の起動	95	その他の情報として、'Unlocking admin account' が保存される
	LDAP ステータス変更に対する変更	123	キーワードまたはオプション：プライマリ URL ステートとセカンダリ URL ステート • 以前の値 • 現在の値 追加情報には、ステータス変更が自動的に行われたか、手動入力されたコマンドによるものかが示されています。
	システムまたは sp_passwordpolicy による、ネットワーク・パスワードの暗号化のための非対称キーペアの再生成	117	extrainfo の情報
select	テーブルからの select	62	キーワードまたはオプション： select into select readtext
	ビューからの select	63	キーワードまたはオプション： select into select readtext
setuser	setuser	84	その他の情報：設定されたユーザ名
table_access	delete	18	キーワードまたはオプション：delete
	insert	41	キーワードまたはオプション：insert
	select	62	キーワードまたはオプション： select into select readtext
	update	70	キーワードまたはオプション： 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 の情報

監査オプション	監査されるコマンドまたはアクセス	イベント	extrainfo の情報
update	テーブルの update	70	キーワードまたはオプション： update writetext
	ビューの update	71	キーワードまたはオプション： update writetext
view_access	delete	19	キーワードまたはオプション：delete
	insert	42	キーワードまたはオプション：insert
	select	63	キーワードまたはオプション： select into select readtext
	update	71	キーワードまたはオプション： update writetext

表 8-6 は、event カラムに表示される値を監査イベント順にリストにしたものです。

表 8-6: 監査イベント値

監査イベント ID	コマンド名	監査イベント ID	コマンド名
1	ad hoc audit record	62	select table
2	alter database	63	select view
3	alter table	64	truncate table
4	bcp in	65	予約済み
5	予約済み	66	予約済み
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	予約済み
12	create trigger	73	監査の有効化
13	create rule	74	監査の無効化
14	create default	75	予約済み
15	create message	76	SSO が変更したパスワード
16	create view	77	予約済み
17	access to database	78	予約済み
18	delete table	79	予約済み
19	delete view	80	役割チェックの実行
20	disk init	81	dbcc

監査イベント ID	コマンド名	監査イベント ID	コマンド名
21	disk refit	82	config
22	disk reinit	83	online database
23	disk mirror	84	setuser コマンド
24	disk unmirror	85	UDR コマンド
25	disk remirror	86	組み込み関数
26	drop database	87	ディスクの解放
27	drop table	88	set SSA コマンド
28	drop procedure	89	kill コマンドまたは terminate コマンド
29	drop trigger	90	connect
30	drop rule	91	reference
31	drop default	92	コマンド・テキスト
32	drop message	93	JCS install コマンド
33	drop view	94	JCS remove コマンド
34	dump database	95	管理者アカウントのロック解除
35	dump transaction	96	quiesce database
36	致命的なエラー	97	create SQLJ 関数
37	致命的ではないエラー	98	drop SQLJ 関数
38	ストアド・プロシージャの実行	99	SSL 管理
39	トリガの実行	100	disk resize
40	grant	101	mount database
41	insert table	102	unmount database
42	insert view	103	login コマンド
43	load database	104	create index
44	load transaction	105	drop index
45	login	106	sp_encryption (暗号化された列の管理)
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	ログインがロックされている
51	サーバのシャットダウン	113	quiesce hold security
52	予約済み	114	quiesce release
53	予約済み	115	パスワード管理
54	予約済み	116	create manifest file
55	役割のオンとオフ	117	regenerate keypair

監査イベント ID	コマンド名	監査イベント ID	コマンド名
56	予約済み	118	alter encryptin key modify encryption
57	予約済み	119	alter encryption key add encryption
58	予約済み	120	alter encryption key drop encryption
59	予約済み	121	alter encryption key modify owner
60	予約済み	122	alter encryption key for key recovery
61	監査テーブルへのアクセス	123	LDAP ステータス変更
		124	alter...modify owner
		127	エラー・ログの管理
		136	transfer table

失敗したログイン試行のモニタリング

ログイン試行の失敗回数が所定の限度を超えたためにログイン・アカウントがロックされると、監査オプションの login_locked と Locked Login (値 112) イベントが記録されます。このイベントは監査オプションの login_locked が設定されると有効になります。login_locked を設定するには、次のように入力します。

```
sp_audit "login_locked","all","all","ON"
```

監査テーブルが満杯でイベントを記録できない場合は、その情報がエラー・ログに記録されます。

ホスト名とネットワークの IP アドレスが監査レコードに記録されます。監査ログを使用して Locked Login イベント (数値 112) をモニタリングすると、ログイン・アカウントに対する攻撃の識別に役立ちます。

ログイン失敗の監査

クライアント・アプリケーションはさまざまな理由でログインに失敗することがありますが、Adaptive Server では、ログイン失敗に関する詳細な情報を提供しません。これは、パスワードの解読や Adaptive Server の認証メカニズムの侵害を意図している悪意のあるユーザに情報を与えることを避けるためです。

ただし、詳細情報は、システム管理者にとっては Adaptive Server の管理上の問題や設定上の問題を診断するために、セキュリティ担当者にとってはセキュリティの侵害を調査するために役に立ちます。

次のように指定することで、すべてのログイン失敗を監査できます。

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

情報の不正使用を防止するために、SSO 役割を付与されたユーザだけが、この機密情報を含む監査証跡情報にアクセスできます。

Adaptive Server は、次の条件に該当するログイン失敗を監査します。

- Windows サービスとして起動された Adaptive Server で、Sybase SQLServer サービスが一時停止された (たとえば Microsoft Management Console for Services によって停止された)。
- リモート・サーバがサーバ対サーバ RPC 用のサイト・ハンドラを確立しようとしたが、リソース不足のため (またはその他の理由で) サイト・ハンドラを初期化できなかった。
- Windows 版の Adaptive Server を trusted ログインまたは統一化ログインを設定して使用しようとしたが、指定されたユーザが信頼された管理者ではなかった (認証できなかった)。
- Adaptive Server が、クライアントによって要求された SQL インタフェースをサポートしていない。
- Adaptive Server がシングルユーザ・モードで稼動してるときにユーザがログインしようとした。シングルユーザ・モードでは、sa_role が付与されているユーザが 1 人だけ Adaptive Server にアクセスできます。sa_role を持っているユーザであっても、追加ログインはできません。
- master データベース内の syslogins テーブルが開かない。これは、master データベースに内部エラーがあることを示します。
- クライアントがリモート・ログインしようとしたが、sysremotelogins が開かない。または、指定されたユーザ・アカウント用のエントリがなく、ゲスト・アカウントも存在しない。
- クライアントがリモート・ログインしようとしたが、指定されたユーザの sysremotelogins 内のエントリがローカル・アカウントを参照しているにもかかわらず、参照先のローカル・アカウントが存在しない。
- クライアント・プログラムがセキュリティ・セッション (Kerberos 認証など) を要求しているが、次の理由でセキュリティ・セッションを確立できない。
  - Adaptive Server のセキュリティ・サブシステムが起動時に初期化されなかった。
  - 構造体に割り当てるメモリ・リソースが不足している。
  - 認証のネゴシエーションが失敗した。
- 指定されたユーザに対して実行される認証メカニズムが見つからない。
- 指定されたパスワードが正しくなかった。
- 指定されたログインで必要なエントリが syslogins に含まれていない。
- ログイン・アカウントがロックされている。
- Adaptive Server のユーザ接続数が制限値に達した。

- unified login required パラメータが設定されているが、適切なセキュリティ・サブシステムによってログインが認証されていない。
- Adaptive Server のネットワーク・バッファを使用できない、または要求されたパケット・サイズが無効である。
- クライアント・アプリケーションがホスト・ベースの通信ソケット接続を要求しているが、ホスト・ベースの通信バッファ用にメモリ・リソースを使用できない。
- シャットダウンが進行中だが、指定されたユーザは SA 役割を持っていない。
- Adaptive Server がログイン用のデフォルト・データベースを開くことができなかった。かつ、このログインには master データベースへのアクセス権がない。
- クライアントは高可用性ログイン・フェールオーバを要求しているが、高可用性サブシステムがこのログインに対して高可用性セッションを確立していない、またはフェールオーバが完了するまでログインが待機できない。
- クライアントは高可用性ログイン設定を要求しているが、高可用性サブシステムがセッションを確立できない、または高可用性セッションのための TDS プロトコル・ネゴシエーションを完了できない。
- Adaptive Server が、ログインに対して tempdb を設定できない。
- TDS ログイン・プロトコル・エラーが検出された。

索引

記号

A

B

C

D

E

F

G

I

K

L

M

N

O

P

R

S

U

W

あ

い

え

お

か

き

く

け

こ

さ

し

す

せ

た

ち

つ

て

と

な

に

ね

は

ひ

ふ

へ

ほ

ま

め

も

や

ゆ

ら

り

る

れ

ろ

わ