

セキュリティの管理

Adaptive Server® Enterprise 16.0

ドキュメント ID： DC01813-01-1600-01
改訂： 2014 年 3 月
Copyright © 2014 by SAP AG or an SAP affiliate company. All rights reserved.
このマニュアルの内容を SAP AG の明示的許可を得ずに、いかなる手段によっても、複製、転載することを禁じます。 ここに記載された情報は事前の通知なしに変更されることがあります。
SAP AG およびディストリビュータが販売しているソフトウェア製品には、他のソフトウェアベンダー独自のソフトウェアコンポーネントが含まれているものがあります。 国内製品の仕様は変わることがあります。
これらの資料は SAP AG および関連会社 (SAP グループ) が情報のみを目的として提供するものであり、いかなる種類の表明または保証も行うものではなく、SAP グループはこの資料に関する誤りまたは脱落について責任を負わないものとします。 SAP グループの製品およびサービスに関する保証は、かかる製品およびサービスに付属している明確な保証文書がある場合、そこで明記されている保証に限定されます。 ここに記載されているいかなる内容も、追加保証を構成するものとして解釈されるものではありません。
ここに記載された SAP および他の SAP 製品とサービス、ならびに対応するロゴは、ドイツおよび他の国における SAP AG の商標または登録商標です。 その他の商標に関する情報および通知については、http://www.sap.com/corporate-en/legal/copyright/index.epx#trademark を参照してください。

目次

第 1 章：セキュリティの概要 1

情報セキュリティの概要 1

Common Criteria 設定評価 2

FIPS 140-2 検証済み暗号化モジュール 3

第 2 章：SAP ASE でのセキュリティ管理 5

セキュリティの設定に関する推奨事項 6

セキュリティの設定例 7

SAP ASE でのセキュリティ機能 8

識別と認証 9

任意アクセス制御 9

役割の分担 10

責任範囲の明確化のための監査 11

データの機密保持 12

第 3 章：SAP ASE のログインおよびデータベースユーザの管理 13

ログインの概要 14

ログインアカウントの作成 14

最後のログインと非アクティブアカウントの管理 15

stale period の定義 16

前回のログインの追跡 16

非アクティブアカウントのロックの防止 16

ログインの認証メカニズム 17

ログインアカウントの変更 17

ログインアカウントの削除 18

パスワードの選択と作成 18
ログインの最大試行回数 19
サーバ全体での最大ログイン失敗回数の設定 19
最大ログイン失敗回数の設定 19
役割の最大ログイン失敗回数の設定 20
ログインの最大ログイン失敗回数の変更 20
役割の最大ログイン失敗回数の変更 20
パスワードが失われた場合のログイン 20
ログインのパスワード情報の表示 21
役割のパスワード情報の表示 22
パスワードが 1 文字以上あるかどうかの検査 22
パスワードの最小長の設定および変更 23
特定のログインの最小パスワード長の設定 23
特定の役割の最小パスワード長の設定 23
特定のログインの最小パスワード長の変更 24
特定の役割の最小パスワード長の変更 24
特定のログインの最小パスワード長の解除 24
複雑なパスワードチェック 25
複雑なパスワードチェックの有効化または無効化 25
単純なパスワードの禁止 25
カスタムの複雑なパスワードチェック 25
パスワードの文字数の指定 26
複雑なパスワードオプションのクロスチェック 27
複雑なパスワードチェックの設定 28
カスタムのパスワードチェックの有効化 30
パスワードのログインと役割の有効期間の設定 33
パスワードによる保護の迂回 34
新規ログインのパスワードの有効期間の作成 35
新規役割のパスワードの有効期間の作成 35
パスワードの作成日の刻印 35

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 35
ネットワーク上でのログインパスワードの保護 36
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 39
パスワード文字セットの考慮事項 39
アップグレードの動作 40
ログインパスワードのダウングレード 41
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 44
アップグレード前の領域の追加 45
ダウングレード 45
allow password downgrade を 0 に設定したときパスワードを期限切れにする方法 48
allow password downgrade の現在の設定値を表示する方法 49
高可用性環境のパスワード 50
アップグレード後に更新されたパスワード 51
パスワードとログインポリシーの設定 51
ログインの失敗 52
ログインアカウントおよび役割のロック 52
ログインと役割のロックとロック解除 53
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 54
ロックされた役割の追跡 56
役割のロックとロック解除 57
スレッショルドを所有するログインのロック 57
ログインプロファイルの管理 57
ログインプロファイルの属性 58
ログインプロファイルの属性の優先度 59
ログインプロファイルの作成 59

デフォルトログインプロファイルの作成 60
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 60
ログインプロファイルの無視 60
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 61
ログインプロファイルの手動での複写 61
ログインプロファイルへの役割の付与 61
ログインスクリプトの起動 62
ログインプロファイル情報の表示 62
ログインプロファイルの修正 63
自動的にアクティブ化された役割の追加または削除 64
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 64
ログインプロファイルの削除 65
データベースへのユーザの追加 65
“guest” ユーザのデータベースへの追加 67
“guest” ユーザのパーミッション 67
ユーザデータベースの “guest” ユーザ 67
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 68
pubs2 と pubs3 の “guest” ユーザ 68
guest ユーザのサーバへの追加 68
リモートユーザの追加 69
グループの作成 69
グループメンバシップの変更 70
グループの作成とユーザの追加 70
データベースのエイリアス 71
エイリアスの追加 72
エイリアスの削除 73
エイリアスに関する情報 73

ユーザ情報を取得する方法 74
ユーザとプロセスのレポート 74
ログインアカウントに関する情報の取得 75
データベースユーザ情報を取得する方法 76
ユーザの名前と ID を表示する方法 76
ユーザ情報の変更 78
パスワードの変更 78
新しいパスワードの要求 79
null パスワード 79
パスワードが失われた場合のログイン 79
ユーザセッション情報の変更 80
ユーザおよびグループの削除 81
ライセンス使用状況のモニタリング 82
ライセンスがカウントされる仕組み 83
License Use Monitor の設定 83
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 83
ユーザライセンス数のロギング 84
ユーザ ID とログイン ID の数 85
ログイン接続の制限 85
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 87
アカウンティング統計を追加する間隔の指定 87

第 4 章：役割の管理 89
システム定義の役割の概要 89
システム標準の役割 90
ユーザ定義の役割の計画 94
ユーザ定義の役割の作成 95
役割のパスワードの追加または削除 95
役割の階層と相互排他性 96

役割の相互排他性の定義と変更 96
役割の階層の定義と変更 97
ログイン時のデフォルトアクティブ化 99
役割アクティブ化の条件 100
ユーザ定義の役割の削除 100
役割のアクティブ化と非アクティブ化 100
役割に関する情報の表示 101
役割 ID と役割名の表示 102
アクティブなシステム標準の役割の表示 102
役割の階層の表示 103
階層内のユーザ定義の役割の表示 103
相互排他性の判別 103
役割のアクティブ化の判別 103
ストアドプロシージャ内の役割の検査 104
役割の付与または取り消し 104
役割の付与 105
grant コマンドと役割について 105
役割の取り消し 106
役割のパスワードのセキュリティ保護 106
文字セットについての考慮事項 107
役割のロックと syssrvroles 107
役割パスワードに適用されるログインパスワードポリシーの確認 108
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 108
SAP ASE での役割の設定 109
SAP ASE のアップグレード 110
SAP ASE のダウングレード 110

第 5 章：外部認証 115
SAP ASE でのネットワークベースのセキュリティの設定 115

SAP ASE のセキュリティサービス 116
ネットワークベースのセキュリティの管理 117
セキュリティの設定ファイル 117
サーバのセキュリティ情報 118
ネットワークベースのセキュリティを使用するための libtcl.cfg の準備 119
objectid.dat ファイル 122
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 123
SAP ASE でのセキュリティの設定 123
ネットワークセキュリティの有効化 123
ユーザに対する認証の要求 124
セキュアデフォルトログインの確立 124
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 125
暗号化によるメッセージの機密保持の要求 126
データ整合性の要求 126
ネットワークベースセキュリティのメモリ要件 126
統一化ログインをサポートするためのログインの追加 127
ログインを追加するための一般的な手順 127
リモート接続での Kerberos セキュリティの確立 128
統一化ログインとリモートサーバログイン 129
リモートサーバ情報の取得 129
セキュリティサービスによるサーバへの接続 129
クライアントのセキュリティメカニズム 130
使用できるセキュリティサービスの情報の取得 131
Kerberos 132
Kerberos の互換性 132
Kerberos 環境での SAP ASE サーバの起動 133
Kerberos の設定 134

プリンシパル名 137
サーバのプリンシパル名 138
sybmapname を使用したユーザプリンシパル名の処理 139
Kerberos による同時認証 142
LDAP ユーザ認証のための SAP ASE の設定 143
生成 DN アルゴリズム 144
検索 DN アルゴリズム 144
新しい SAP ASE での LDAP の設定 145
既存の SAP ASE サーバの LDAP へのマイグレーション 145
LDAP ユーザ認証の管理 146
LDAP ユーザ認証パスワード情報の変更 148
フェールオーバのサポート 149
ログインと LDAP ユーザアカウント 149
セカンダリ検索サーバのサポート 150
LDAP サーバのステータスの移行 151
LDAP ユーザ認証のチューニング 153
ログインマッピングに対する制御の強化 154
外部認証のログインマッピング 156
LDAP ユーザ認証エラーのトラブルシューティング ..157
LDAP サーバの設定 158
LDAPS ユーザ認証 159
自動的な LDAP ユーザ認証とフェールバック 159
LDAP フェールバック時間間隔 160
PAM を使用する認証のための SAP ASE の設定 161
SAP ASE での PCI の有効化 162
オペレーティングシステムの設定 162
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 162
PAM ユーザ認証のための SAP ASE の設定 163
ログインと PAM ユーザアカウント 163

機能拡張されたログイン制御 164
認証の強制 164
sp_maplogin を使用したログインのマップ 165
マッピング情報の表示 165
認証メカニズムの設定 166

第 6 章：ユーザパーミッションの管理 167
データベース作成用のパーミッションの付与 168
データベース所有権の変更 169
データベース所有者の権限 169
データベースオブジェクト所有者の権限 171
システムプロシージャに対するパーミッション 172
パーミッションの付与または取り消し 172
オブジェクトアクセスパーミッション 173
具体的 ID 174
SQL92 標準に準拠するための要件 175
オブジェクトアクセスパーミッションの付与の例 175
オブジェクトアクセスパーミッションの取り消しの例 176
dbcc コマンドのパーミッションの付与 177
サーバワイドな dbcc コマンドとデータベース固有の dbcc コマンド 177
システムテーブルのパーミッション 178
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 178
grant 文と revoke 文の組み合わせ 180
パーミッションの順序と階層 181
別のユーザのパーミッションの取得 182
setuser コマンド 183
代理権限 183
set proxy を使用した役割の制限 184

代理権限の実行 185
アプリケーションによる代理権限の使用方法 ... 187
データベースオブジェクトの所有権の変更 187
サポートしているオブジェクトタイプ 187
所有権を譲渡する権限 188
所有権の譲渡 189
システムデータベースのオブジェクトの所有権の譲渡 189
データベース所有者のオブジェクトの所有権の譲渡 190
preserve permissions コマンド 190
セキュリティに関する情報 191
暗号化キーの所有権の譲渡 191
パーミッションのレポート 192
sysprotects テーブルに対する代理権限に関するクエリの実行 192
ユーザとプロセスに関する情報の表示 192
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のレポート 193
特定のテーブルに対するパーミッションのレポート 194
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 195
セキュリティメカニズムとしてのビューの使用 195
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 197
役割とストアドプロシージャ 198
所有権の連鎖の理解 199
ビューと所有権の連鎖の例 200
プロシージャと所有権の連鎖の例 201
トリガのパーミッション 202

execute as owner または execute as caller を含むプロシージャの実行 202
修飾されていない名前を使用してオブジェクトを参照するプロシージャの作成 206
完全修飾名を使用して別のデータベースでネストされたプロシージャを呼び出すプロシージャ 208
ローレベルアクセス制御 209
アクセスルール 210
アクセスルールの構文 211
アクセスルールと拡張アクセスルールの例 213
アクセスルールと alter table コマンド 215
アクセスルールと bcp コマンド 215
ユーザ定義 Java 関数としてのアクセスルール 215
Application Context Facility の使用 218
アプリケーションコンテキスト関数によるパーミッションの設定 218
SYS_SESSION システムアプリケーションコンテキスト 220
アクセスルールと ACF による問題の解決 221
ログイントリガの使用 222
ログイントリガの作成 223
ログイントリガの設定 224
ログイントリガの実行 225
ログイントリガの出力 225
その他のアプリケーションでのログイントリガの使用 226
ログイントリガの制限事項 229
ログイントリガに対する execute 権限の無効化 230
ログイントリガからの set オプションのエクスポート 230

グローバルログイントリガの設定 231

第 7 章：述部付きの権限 233

アクセスルールと述部付きの権限の違い 233
述部付きの権限に使用するコマンド 234
述部付きの権限を使用するための SAP ASE の設定 234
enable predicated privileges 235
述部付きの権限の付与 235
データを選択するためのアクセス権の付与 236
データを更新するためのアクセス権の付与 237
データを削除するためのアクセス権の付与 237
述部付きの権限を使用したデータプライバシーポリシーの実施 237
述部付きの権限の取り消し 238
SAP ASE における述部付きの権限の sysprotects への保存の仕組み 239
述部付きの役割のアクティブ化 240
ローレベル権限を適用するための述部の結合 241
例 1: 同じテーブルアクセスについての複数の付与の述部の結合 242
例 2: さまざまなカラムセットに適用される複数の付与の述部の結合 243
例 3: 異なるアクセス権についての複数の付与の述部の結合 244
述部付きの権限がある場合の SQL の動作 245
述部付きの権限に対する所有権の連鎖の影響 246
ansi_permissions と述部付きの権限 246
述部によるアクセスのパーミッション 248
述部付きの権限とトリガ 249
述部付きの権限の再コンパイル 249
述部の再帰的処理の禁止 250
述部による情報の漏洩 251

第 8 章：細密なパーミッション 253
細密なパーミッションを使用するための SAP ASE の設定 253
システム権限 254
システム標準の役割に含まれる権限の影響 254
パーミッションの管理 255
manage security permissions 権限 255
manage server permissions 権限 256
manage database permissions 権限 259
manage any object permission 権限 261
システム標準の役割に付与される権限 262
データベース所有者に割り当てられる権限 265
細密なパーミッションによる sa_serverprivs_role の追加 267
システム管理者に付与されるデフォルトの役割 269
システム管理者およびデータベース所有者の権限の制限 ...270
Enable granular permissions と sybsecurity 271
ロックアウトされたサーバへのログイン 271
一般的な使用シナリオ 272
システムテーブル master.dbo.sysprotects 274
データベースユーザ usedb_user アカウント 276
付与可能なシステム権限 277

第 9 章：データの機密保持 305
パブリックキー暗号法 305
SSL の概要 308
SAP ASE での SSL 308
SSL フィルタ 308
証明書による認証 309
接続タイプ 310

SSL の有効化 311
証明書の要求 312
サーバディレクトリエントリの作成 313
証明書の管理 314
パフォーマンス 316
暗号スイート 316
@@ssl_ciphersuite 317
SSL 暗号スイートの優先度 317
SSL を使用した共通名の指定 322
sp_listener での共通名の指定 322
Kerberos による機密保持 323
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 323
パスワードと文字セット 324
残存データの削除 324
残存データが発生する操作 325
残存データの削除の有効化 327
残存データの削除例 328
残存データを削除するための設定の組み合わせ 331
残存データの削除が有効になっているかどうかの確認 332
制限事項 333

第 10 章：監査 335
監査システム 336
sybsecurity データベース 336
監査証跡 336
監査キュー 337
監査システムの管理 338
監査システムのインストール 338
監査デバイスに関するインストール前の推奨事項 339

installsecurity による UNIX への監査のインストール 340
instsecu による Windows への監査のインストール 340
auditinit による監査のインストール 342
監査の有効化と無効化 348
監査の再起動 348
監査システムの設定 348
ソースコードコントロールシステムでの展開されたソースの管理 349
設定の変更の監査 349
SAP ASE 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 350
グローバル監査オプションの設定 350
監査システムの変更 369
sybsecurity 監査データベースの移動 369
auditinit によるデバイスエントリの削除 371
auditinit によるデバイスエントリの変更 372
監査証跡の管理 373
複数の sysaudits テーブルを sybsecurity に作成 373
スレッショルドプロシージャの設定 373
監査設定パラメータ 378
監査証跡の問い合わせ 380
監査証跡へのユーザ指定レコードの追加 380
トランザクションログのトランケート 382
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 382
単一テーブル監査 383
単一テーブル監査のスレッショルドプロシージャ 384

現在の監査テーブルが満杯になったときに起こる動作 385
現在の監査テーブルが満杯になったときのリカバリ 385
監査テーブル 386
extrainfo カラムの読み込み 387
失敗したログイン試行のモニタリング 402
ログイン失敗の監査 402

第 1 章　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。

情報セキュリティの概要

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。

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します。
- 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、理解します。
- 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ります。

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。

Common Criteria 設定評価

SAP® Adaptive Server® Enterprise は、CCEVS (Common Criteria Evaluation and Validation Scheme) の規定に従って評価されました。 また、SAP ASE® 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。

Common Criteria for Information Technology 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。

SAP ASE 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バージョン 15.0.1 で構成されます。 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。 SAP ASE 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、SAP ASE 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。

SAP ASE 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 - SAP ASE 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 - SAP ASE 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、役割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 - SAP ASE 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、SAP ASE 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。 SAP ASE 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。

SAP ASE 用の SSL は、SSL および TLS プロトコルとともに暗号ライブラリを実装するオープンソースツールキット OpenSSL をサポートします。 詳細については、Open Source を参照してください。

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。 FIPS ログインパスワードの暗号化が有効な場合、すべての暗号化操作で OpenSSL FIP 140-2 が使用されます。 無効になっている場合は、OpenSSL の FIPS 以外が使用されます。

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： 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 SAP ASE でのセキュリティ管理

SAP ASE を安全に管理するための主要タスクを実行します。

1. 監査機能を含む、SAP ASE 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「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。 「SAP ASE 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。 「SAP ASE のログインおよびデータベースユーザの管理」と「ユーザ定義の役割の作成」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。 「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、SAP ASE 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。 『TransactSQL 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、SAP ASE 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。「監査」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。「外部認証」および「データの機密保持」を参照してください。

参照：

- 第 10 章、「監査」 (335 ページ)
- 第 3 章、「SAP ASE のログインおよびデータベースユーザの管理」 (13 ページ)
- ユーザ定義の役割の作成 (95 ページ)
- 第 6 章、「ユーザパーミッションの管理」 (167 ページ)
- 第 5 章、「外部認証」 (115 ページ)
- 第 9 章、「データの機密保持」 (305 ページ)

セキュリティの設定に関する推奨事項

ログインの設定と監査のためのセキュリティに関する推奨事項

- sa ログインの使用 - SAP ASE 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！ SAP ASE 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 - 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、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

安全な操作環境を設定するには、セキュリティの役割と特定のコマンドを使用します。

次の表は、ユーザに割り当てられる特別な役割を示します。

表 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

この表は、上記に示した役割の割り当てに基づいて SAP ASE 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 : 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 SAP ASE 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”‘**	サーバワイドな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、監査を有効化してください。	
• **create login**	ログインとパスワードを追加する。

コマンド	結果
• **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のみを使用できる。
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” ログインをロックしてください。	

SAP ASE でのセキュリティ機能

SAP ASE には機密データの保護に役立つセキュリティ機能があります。

- 識別と認証 - 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します。 SAP ASE 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ます。
- 任意アクセス制御 - 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します。通常は **grant** コマンドと **revoke** コマンドを使用します。 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ます。
- 役割の分担 - 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します。 SAP ASE 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ります。 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ます。
- 責任範囲 - 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できるようにします。 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ます。
- データの機密保持- 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します。 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します。 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ます。

識別と認証

内部認証メカニズムと外部認証メカニズムの両方を使用できます。

SAP ASE 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、このサーバユーザ 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。

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。 SAP ASE 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でオペレーティングシステムインタフェースを使用する、集中化された認証メカニズムを提供します。

リモートサーバの管理
サーバ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「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

任意アクセス制御を使用すると、ユーザ、グループ、役割にさまざまなパーミッションを付与することができます。

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。 ユーザ、グループ、役割に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。 パーミッションを無効にするには、**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。

ローレベルアクセス制御
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、インストール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

述部付きの権限
述部付きの権限を使用すると、テーブル所有者は、**grant** 文で SQL **where** を指定することによって、ユーザ、グループ、または役割にローレベルのアクセス権を提供できます。 他のテーブルへのアクセス権など、SQL の全機能を使用して包括的なローレベルのセキュリティポリシーを実装します。

役割の分担

SAP ASE 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。 SAP ASE 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、役割の割り当て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、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自動的にそのユーザに与えられるというものです。 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。

役割のアクティブ化の制限
セキュリティ管理者は、**grant role** 文で **where** 句を指定することによって、役割をアクティブ化できる条件を制限します。 ログイン時に自動的に、または **set role** 文の処理時に役割がアクティブ化されるときに、**where** 句で表された条件が評価されます。

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。

責任範囲の明確化のための監査

SAP ASE 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
- すべての監査オプションを設定するための **sp_audit**
- 監査証跡にユーザ定義レコードを追加するための **sp_addauditrecord**

監査機能のインストール時に、SAP ASE 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

データの機密保持

SSL (Secure Socket Layer) 標準や Kerberos を使用してクライアントと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** 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、『暗号化カラムユーザーズガイド』、『システム管理ガイド 第 2 巻』の「ユーザデータベースのバックアップとリストア」を参照してください。

第 3 章

SAP ASE のログインおよびデータベースユーザの管理

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。

注意： 操作のパーミッション要件は、細密なパーミッションが無効であることを前提としています。 細密なパーミッションが有効な場合は、操作が異なることがあります。

表 3 : SAP ASE でのユーザの管理

作業	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ある。	**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

作業	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
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

参照：

- 第 8 章、「細密なパーミッション」 (253 ページ)

ログインの概要

ログインは、ユーザの名前とパスワードを定義して、SAP ASE へのアクセスを可能にします。

create login を実行すると、ローがが `master.dbo.syslogins` に追加され、新しいユーザに一意のシステムユーザ ID (`suid`) が割り当てられ、指定された属性情報が入力されます。ユーザがログインするときに、SAP ASE はそのユーザが指定した名前とパスワードを `syslogins` の中で見つけます。 `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作成

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。

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特定のパーミッションを、ユーザまたはグループに付与します。

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』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、サーバ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。 SAP ASE 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割り当てられる連続した整数値です。

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” はサーバ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

SAP ASE アカウントを管理し追跡します。

SAP ASE 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、非アクティブであることを理由としたロックがログインアカウントに適用されていない場合、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かどうかを確認するために、**syslogins.lastlogindate** フィールドと **syslogins.pwdate** フィールドがチェックされます。非アクティブであるためにログインアカウントがロックされると、`syslogins` 内の `locksuid` フィールド、`lockreason` フィールド、`lockdate` フィール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** 属性で設定できます。

```
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、SAP　ASE によって定義済みの外部認証メカニズムがあるかどうかが調べられます。 定義済みの外部認証メカニズムが検出された場合は、それが使用され、それ以外の場合は、**ASE** メカニズムが使用されます。

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すること、ログイン試行の失敗回数を制限すること、属性を削除すること、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することができます。

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、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。 SAP 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

参照：
- 複雑なパスワードチェック (25 ページ)

ログインの最大試行回数

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。

サーバ全体での最大ログイン失敗回数の設定

sp_passwordpolicy を使用して、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に設定します。

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。
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。

最大ログイン失敗回数の設定

特定のログインの作成時に、そのログインの最大失敗回数を設定します。
特定のログインの作成時に、そのログインの最大ログイン試行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。

役割の最大ログイン失敗回数の設定

特定の役割の作成時に、その役割の最大ログイン失敗回数を設定します。
特定の役割の作成時に、その役割の最大ログイン失敗回数を設定するには、**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” に対する最大ログイン失敗回数を 20 に設定します。

```
create role intern_role with passwd "temp244", max 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。

ログインの最大ログイン失敗回数の変更

特定のログインの最大ログイン失敗回数を変更します。
既存のログインの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。

次の例では、ログイン “joe” の最大ログイン失敗回数を 40 に変更します。

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

役割の最大ログイン失敗回数の変更

特定の役割の最大ログイン失敗回数を変更します。
既存の役割の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。

次の例では、“physician_role” の最大ログイン失敗回数を 5 に変更します。

```
alter role physician_role set max failed_logins 5
```

次の例では、すべての役割の最大ログイン失敗回数を無効にする設定を削除します。

```
alter role "all overrides" set max failed_logins -1
```

max 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 を使用して、sa_role と sso_role のパスワードを再設定できます。

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、SAP ASE

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。

sa_role のパスワードがない場合は、-p sa_role を使用して起動すると、エラーログにエラーメッセージが出力されます。

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。

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。

```
xml:space="preserve">dataserver -p sa_role
```

次のメッセージが表示されます。

```
xml:space="preserve">New password for role 'sa_role' :
qjcdyrbfkxgyc0
```

ログインのパスワード情報の表示

sp_displaylogin を使用して、ログインのパスワード情報を表示します。

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、次のように入力します。

```
sp_displaylogin [loginame [, expand_up | expand_down]]
```

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。

役割のパスワード情報の表示

sp_displayroles を使用して、ログインのパスワード情報を表示します。
役割のログイン設定とパスワード設定を表示するには、次のように入力します。

```
sp_displayroles
      [grantee_name [, mode]]
```

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。

- 1 文字以上あるかどうかの検査の有効化
  パスワードの検査機能をアクティブにするには、次のように入力します。
  ```
  sp_configure "check password for digit", 1
  ```
- 1 文字以上あるかどうかの検査の無効化
  パスワードの検査機能を非アクティブにするには、次のように入力します。
  ```
  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。

パスワードの最小長の設定および変更

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： SAP ASE は、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最小パスワード長
- ログインごとまたは役割ごとの最小パスワード長

ログインごとまたは役割ごとの最小パスワード長は、サーバワイドの設定よりも優先されます。 最小パスワード長の設定は、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。

特定のログインの最小パスワード長の設定

create login を使用してログインを作成するときに、そのログインの最小パスワード長を設定します。
次に例を示します。

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” の最小パスワード長を 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。

特定の役割の最小パスワード長の設定

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** を設定します。
次に例を示します。

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、“intern_role” の最小パスワード長 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、**min passwd 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。

特定のログインの最小パスワード長の変更

既存のログインの最小パスワード長を設定または変更するには、**alter login** を使用します。
次に例を示します。

次の例では、ログイン “joe” の最小パスワード長を 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。

特定の役割の最小パスワード長の変更

既存の役割の最小パスワード長を設定または変更するには、**alter role** を使用します。
次に例を示します。

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、すべての役割の最小パスワード長を無効にします。

```
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。

特定のログインの最小パスワード長の解除

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。
次に例を示します。
次の例では、ログイン joe の最小パスワード長の制限を解除します。

```
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

sp_passwordpolicy を使用して、複雑なパスワードチェックを有効にします。

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。

複雑なパスワードチェックの有効化または無効化

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。

- 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'
  ```

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

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するには、**disallow simple password** を使用します。

disallow simple password を次のいずれかに設定できます。

- 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

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、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。

パスワードの文字数の指定

sp_passwordpolicy パラメータを使用して、パスワード中の最小文字数 (数字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 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許可される最大ログイン失敗回数。 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クロスチェック

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必要があります。
- **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必要があります。
- **systemwide password expiration** には、**password exp warn interval** よりも大きい値を設定する必要があります。

上記の相互チェックを行うために、SAP ASE 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。

SAP ASE 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

ログインレベルの複雑なパスワードオプションの設定には、**create login** または **alter login** を使用します。 グローバルレベルのオプションの設定には、**sp_passwordpolicy** または **sp_configure** を使用します。

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

グローバル単位およびログイン単位で、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 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'
```

パスワードは次の順序でチェックされます。

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すで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すでにチェックされているため、不要です。

指定された順序でチェックされ、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、SAP ASE 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
```

このチェックは失敗し、エラーメッセージが表示されます。 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、パスワードの最小文字数を 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
```

これらのチェックは次の順序で行われます。

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。

パスワードは変更されずに、エラーメッセージが表示されます。

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

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。

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
```

新しいストアドプロシージャ **master.dbo.sp_extrapwdchecks** を作成します。このストアドプロシージャでは、以前使用したパスワードを暗号化形式で pwdhistory テーブルに保存し、使用済みのパスワードの再利用を禁止します。 **create login** コマンドまたは **alter login … modify password** コマンドが実行されると、ユーザ定義

ストアドプロシージャ **sp_extrapwdchecks** が自動的に呼び出されます。 以下に **sp_extrapwdchecks** の実装の例を示します。

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

begin
declare @current_time datetime,
        @encrypted_old_pwd varbinary(30),
        @encrypted_new_pwd varbinary(30),
        @salt varchar(8),
        @changedby varchar(30),
        @cutoffdate datetime

        select @changedby = suser_name()
        select @salt = null

       -- NOTE : caller_password and/or loginame arguments can be null.
        -- In these cases, password history checks should be skipped.

        -- @loginame is null when SSO creates a new login account
        -- using “create login” command.

        -- @caller_password is null when

        -- 1. SSO creates a new login account using
        -- “create login” command.

        -- 2. SSO modifies the login account’s password using
        -- “alter login … modify password” command.

        -- Business logic for custom password checks should be
        -- implemented here.

        -- If there is no need to maintain password history, return
        -- from here.

        if (@loginame is NULL)
        begin
            return 0
        end

       -- Change this line according to the needs of your installation.

        -- This checks below keep history of 12 months only.

        select @current_time = getdate(),@cutoffdate = dateadd
           (month, -12, getdate())

        delete master..pwdhistory
```

```
        where name = @loginame
        and pwdate < @cutoffdate

       select @salt = substring(password, 1, 8) from master..pwdhistory
            where pwdate =
                (select max(pwdate) from master..pwdhistory where
                name=@loginame)and name=@loginame

        if @salt is null
        begin

            select @salt = substring(hash
                (password_random(8), 'sha1'), 1, 8)
        end

        select @encrypted_new_pwd = @salt + hash
                (@salt + @new_password, 'sha1')

        if not exists ( select 1 from master..pwdhistory
           where name = @loginame and password = @encrypted_new_pwd )


        begin
            -- new password has not been used before

            if (@loginame != @changedby)
            begin
                return 0
            end

            -- Save old password
            select @encrypted_old_pwd = @salt + hash
                (@salt + @caller_password, 'sha1')
            insert master..pwdhistory
                select @loginame,
                    @encrypted_old_pwd,@current_time, @changedby
            return (0)
        end
        else
        begin
               raiserror 22001 --user defined error message
        end

    end
    go
```

ユーザ定義メッセージ 22001 を追加するには、**sp_addmessage** を使用します。`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、これらのプロシージャ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

システム管理者およびシステムセキュリティ担当者は、ログインまたは役割の作成時にログインまたは役割の有効期間を設定できます。また、既存のログインまたは役割の有効期間を変更できます。

コマンド	有効期間のオプション
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)。
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)。

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ます。 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ます。
- 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りません。
- 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。

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 はサーバにログインし、`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 30 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 7 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 30 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

RSA パブリックキー暗号化アルゴリズムを使用して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には、非対称暗号化を使用できます。

SAP ASE 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します。 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します。 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します。

SAP ASE 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。 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ます。

sp_configure のパラメータ **net password encryption reqd** は 3 という値をサポートしており、この値は、このサーバでは、EPEP ログインプロトコルを使用している着信クライアントのみが許可されることが示されます。 この値が 0、1、および 2 の場合も、EPEP ログインプロトコルをサポートするクライアントが、EPEP ログインプロトコルを実装した SAP ASE と共に使用しようとしたときに、EPEP ログインプロトコルを使用できます。

この値を 2 または 3 に設定すると、このプロトコルを使用した最大設定接続をサポートするために、ネットワークメモリが増やされます。 **additional network memory** 設定パラメータは、EPEP によって使用されるネットワークメモリプールに、より多くのメモリを動的に追加します。 **net password encryption reqd** の値を 3 に設定すると、KPP ハンドラは sleep ステータスになります。これは、すべての接続に新規の RSA キーペアを提供する必要がないためです。 つまり、各接続に対してユニークな RSA キーペアを生成するために CPU サイクルを消費することが回避されるため、CPU をより効率的に使用できます。

KPP ハンドラのステータスを調べるには、**sp_who** コマンドを使用できます。

『システム管理ガイド 第 1 巻』の「設定パラメータ」の「net password encryption required」を参照してください。

RSA 非対称暗号化を使用する 2 つのバージョンのログインプロトコルがサポートされています。 一方はログインセッションごとにユニークなキーペアを保証し、もう一方はログインプロトコルでランダムな番号を使用します。 ネットワークパスワードの暗号化を使用するユーザ接続が多数ある場合、セッションごとにユ

ニークなキーペアを生成すると、新しいキーペアの計算が原因で計算能力がピークに達することがあります。 2 番目のアプローチの方が、必要な計算能力は少なくて済みますが、 ランダムな番号を生成する RSA 非対称暗号化を使用する新しいログインプロトコルをサポートする、再コンパイルされたクライアントプログラムが必要です。

非対称キーペアの生成
ランダムな番号を使用する RSA 非対称暗号化では、次の場合に新しいキーペアが生成されます。

- サーバ起動時
- 自動的に 24 時間間隔で、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。

ランダムな番号を使用しない RSA 非対称暗号化の場合は、デフォルトで、接続ごとにキーペアが生成されます。

sp_passwordpolicy プロシージャの **unique keypair per session** オプションを使用すると、このログインプロトコルについて接続ごとのキーペアの生成を有効または無効にすることができます。 ただし、ネットワークパスワードのセキュリティを考慮する必要がない環境のみで使用してください。ランダムな番号要素を使用せずに、キーペアが再利用されるためです。

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"
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、パスワードの暗号化を使用するかどうかの判断には、**net password encryption** オプションが使用されます。

このサーバオプションが **true** に設定されていると RSA または独自のアルゴリズムが使用されます。 **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` に保存され、**sp_helpserver** ストアドプロシージャを使用してサーバオプションの値を表示できます。

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、`ASEnterprise` クラス値を持つ `sysservers` エントリの **net password encryption** は、**true** に設定されます。 他のサーバクラスは変更されません。 これによって、SAP ASE サーバ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。

下位互換性

- SAP 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。
- ランダムな番号を生成する RSA アルゴリズムを使用するには、SAP ASE バージョン 15.7 ESD #1 以降と新しい Connectivity SDK クライアント (バージョン 15.7 ESD #1 以降) が必要です。
- RSA アルゴリズムを使用するには、SAP ASE 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。
- SAP 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 SAP 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。
- RSA アルゴリズムも SAP 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、SAP ASE 15.0.2 以降は RSA 暗号化を使用します。 15.0.2 より古い SAP ASE は SAP 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

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 SAP ASE 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、SAP ASE 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。 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。

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するため、クリアテキストのパスワードは標準の形式に変換してから暗号化されます。

パスワードを標準の形式に変換してから `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。

アップグレードの動作

アップグレードするときの master データベースの動作が変わります。

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、暗号化されたパスワードは、`password` カラム内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` カタログに維持されます。

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。
- `password` カラムは、古い暗号化アルゴリズムを使用して更新され、続いて新しい暗号化アルゴリズムを使用して更新されます。

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 `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'
```

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、**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 SAP 独自のアルゴリズムと 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、ユーザパスワードをリセットせずに 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)
```

行 **Login Password Encryption** の値 **SYB-PROP**: **SYB-PROP** は、そのアカウントでは SAP 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。

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、1,000 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

SAP ASE 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、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。

sp_downgrade を実行して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 SAP ASE 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

`master` データベースとトランザクションログには、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、SAP ASE 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、バージョン 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バージョン 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 SAP ASE 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

SAP ASE 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。 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、SAP 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。 SAP ASE 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、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期限切れにする方法

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します。

allow password downgrade 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

valuemessage
-------------------------------------------------------------
1Password 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

valuemessage
-------------------------------------------------------------
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 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

valuemessage
-------------------------------------------------------------
NULLNew master database.
(1 row affected)
```

高可用性環境のパスワード

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 SAP ASE 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。 同様に、以前のバージョンにダウングレードするには、**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

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以下を行えます。

- 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
- ログインの複雑な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

ログインの失敗

データにアクセスするためには、ユーザは認証される必要があります。

認証が失敗した場合は、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。

ログイン失敗の理由は `sysaudits.extrainfo` カラムの `Other Information` の項にある `Errornumber.Severity.State` に示されます。 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 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"
```

ログインアカウントおよび役割のロック

ユーザが SAP ASE にログインできないようにするには、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。

ログインと役割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。

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。
- 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。

構文は次のとおりです。

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

構文の説明は次のとおりです。

- *login_name* は、ロックまたはロック解除されるアカウントの名前です。 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、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。

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。

ロックされたログインの表示
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。

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。

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。

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、ロックされたアカウントに関する情報を追跡するカラムがあります。

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 SAP ASE 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 loginupdates** 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 'all', '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 SAP ASE 内部で処理されます。

表 4 : locksuid の値と理由

lockreason の値	locksuid の値	`lockreason account` 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ロックされた。

lockreason の値	locksuid の値	lockreason account の説明
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。

ロックされた役割の追跡

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
```

これらのカラムは、ロック情報のための syssrvroles 内にあります。

- 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ない。
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た。
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** に達したため役割がロックされた。

役割のロックとロック解除

役割のロックおよびロック解除には、**alter role** を使用します。
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します。

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

スレッショルドは、ロックされたユーザログインの影響を受けます。

- 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。

次の表には、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。

作業	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

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。

次の表には、ログインプロファイルの属性がまとめられています。

属性	説明
default database	デフォルトのデータベース。
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る。
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、エラーが生成される。 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ない。

属性	説明
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値 **ANY** が使用される。
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非アクティブ状態を保持できる期間を示す。この期間を経過すると、非アクティブであることが理由でロックされる。
profile id	データベースを指定する。

ログインプロファイルの属性の優先度

ログインプロファイルの属性は、優先度に基づいてログインアカウントに関連付けられます。

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

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。

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。

デフォルトログインプロファイルの作成

ログインプロファイルは、すべてのログインアカウントのデフォルトとして定義できます。

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。

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、直接関連付けられているログインプロファイルまたは、デフォルトのログインプロファイルから関連付けられているログインプロファイルを無効にするときに使用します。

ログインアカウントの対応する属性は、優先度の規則に従って適用されます。

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

新しいログインプロファイルに既存のログインアカウント値を転送します。

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、SAP ASE 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` を複写 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

ログインプロファイルの作成時にログインプロファイルに役割を付与します。

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。

アクティブ化述部を表す **where** 句を使用してデフォルトの役割がログインに付与されている場合は、アクティブ化述部が true と評価されたときにのみ、デフォルトの役割がアクティブ化されます。

参照：

- 自動的にアクティブ化された役割の追加または削除 (64 ページ)

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。

構文は次のとおりです。

```
create login profile ravi_lp 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。
- **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。

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、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。

参照：

- ログイントリガの使用 (222 ページ)

ログインプロファイル情報の表示

名前、ID、バインド情報など、ログインプロファイルに関する情報を表示します。

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。

この例は、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。

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
```

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』の「**create login profile**」を参照してください。

参照：

- ログインプロファイルの属性 (58 ページ)

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。 役割は、次のようにユーザのログインプロファイルに自動的に付与されます。

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、userGroup_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

コマンド **drop loginprofile** 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、SAP ASE ログインをすで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]]
```

各パラメータの意味は、次のとおりです。

- *loginame* - 既存のユーザのログイン名を指定します。
- *name_in_db* - 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」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* - 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。

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。

参照：

- グループメンバシップの変更 (70 ページ)

“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、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、SAP ASE 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。

`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サーバ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、SAP ASE 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。

“guest” ユーザのパーミッション

SAP ASE 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。

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” の権限よりも拡大または縮小することができます。 “guest” は “public” の権限を継承します。

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

SAP ASE 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、新規ユーザは『TransactSQL 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**、および **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、SAP 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 SAP ASE サーバ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「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、SAP ASE 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。

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。

グループの作成とユーザの追加

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。

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)。

データベースのエイリアス

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。 サーバ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。

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、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。 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。

参照：

- 第 10 章、「監査」 (335 ページ)
- 具体的 ID (174 ページ)

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。

構文は次のとおりです。

```
sp_addalias loginame, name_in_db
```

構文の説明は次のとおりです。

- *loginame* - 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。 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* - *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、`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。

代替 *suid* からユーザ ID へのマッピングを削除すると、`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。

エイリアスに関する情報

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

ユーザ、グループ、および使用状況に関する情報は、次のプロシージャを使用して取得します。

作業	プロシージャ
現在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**

ユーザとプロセスのレポート

sp_who を使用すると、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。

構文は次のとおりです。

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

構文の説明は次のとおりです。

- *loginame* - ユーザのログイン名。 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* - 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 orignam 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。

構文は次のとおりです。

```
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、次のコマンドは sa の役割を表示します。

```
sp_displaylogin 'mylogin'
```

```
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内のユーザ名です。

```
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。

表示対象	使用するコマンド	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。 指定しなかった場合は、現在のユーザに関する情報が表示されます。

次の例は、ユーザ “sandy” のサーバユーザ ID を表示する方法を示します。

```
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()

-----------------------------------
mary4
```

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 5 : user_id システム関数と user_name システム関数

表示対象	使用するコマンド	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。 指定しなかった場合は、現在のユーザに関する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

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コマンドおよびシステムプロシージャについて説明します。

作業	必要な役割	システムプロシージャ	alter/create/drop login/login profile コマンドの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 profilealter 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、すべてのユーザが自分のパスワードをいつでも変更できます。 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、次のように入力します。

```
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。

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。『システム管理ガイド 第 1 巻』の「設定パラメータ」を参照してください。

設定パラメータよりもパスワードポリシーの設定が優先されます。

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インを 2012 年 8 月 1 日の午前 10 時半に作成したとすると、2012 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付が記録されます。 次のクエリは、2012 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 2012"
```

null パスワード

null 以外のパスワードを使用してログインアカウントを保護します。

null パスワードを割り当てることはできません。 ただし、SAP ASE 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

sa および sso のアカウントと役割がロックされた場合やパスワードが失われた場合に新しいパスワードを設定するには、**dataserver** コマンドを使用します。

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、SAP ASE 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、SAP ASE サーバ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]
```

構文の説明は次のとおりです。

- *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の名前です。
- *application_name* - SAP ASE 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 “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、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、任意のクライアント接続の接続情報を表示できます)。 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の接続情報を表示するには (ユーザ “alison” が自分の接続情報を表示する場合など)、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、データベース所有者は、**sp_dropuser** または **sp_dropgroup** を使用して、ユーザおよびグループをデータベースから削除できます。

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。通常、*name_in_db* はログイン名です。ただし、**sp_adduser** を使用して別の名前が割り当てられている場合は除きます。

```
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** を使用して、そのユーザによってパーミッションを付与されたすべてのユーザからパーミッションを取り消した後、その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。

参照：
- グループメンバシップの変更 (70 ページ)

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 SAP ASE 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。

SAP ASE サーバ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 SAP 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、SAP ASE 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。

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です。 あるユーザが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で 1 ライセンスが使用されます。

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、SAP ASE 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、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 設定パラメータと **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「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

SAP ASE 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。 License Use Monitor は 24 時間ごとに、`syblicenseslog` 内のカラムを更新します。

`syblicenseslog` テーブルのカラムを次に示します。

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 2012 11:43AM         123
     0    Jul 18 2012 11:47AM         147
     1    Jul 19 2012 11:51AM         154
     0    Jul 20 2012 11:55AM         142
     0    Jul 21 2012 11:58AM         138
     0    Jul 21 2012  3:14PM         133
```

この例では、2012 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。

SAP ASE 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。 SAP ASE 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。

2012 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。

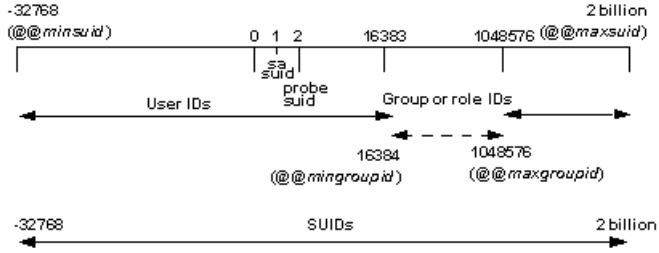
ユーザ ID とログイン ID の数

SAP ASE 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。

次に ID タイプごとの有効な範囲を示します。

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

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

サーバごとに 20 億以上のログインを定義できますが、実際に同時接続が可能なユーザの数は制限されます。

実際のユーザ数は、次の値によって制限されます。

- **number of user connections** 設定パラメータの値。
- 使用可能なファイル記述子の数。 各ログインは接続にファイル記述子を 1 つ使用します。

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
SAP ASE 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。次に、**number of user connections** を 32,000 以上の値に設定します。

注意： 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。 ファイル記述子の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。

次にログイン、ユーザ、グループに関するグローバル変数を示します。

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。 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、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、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、CPU と I/O の統計が累積されます。 **sp_clearstats** を実行すると、ユーザの新しいアカウンティング期間が開始され、**sp_reportstats** が実行されて前の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、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、アカウンティング統計を `syslogins` に追加する頻度を指定できます。

アカウンティング統計を `syslogins` に追加する基準となるマシンのクロックチックの累積数を指定するには、**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、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 です。 最大値は 2,147,483,647 です。

第 4 章　役割の管理

役割とは、役割の割り当て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。 SAP ASE 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。

SAP ASE 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、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。

注意： この章で説明する操作のパーミッション要件は、細密なパーミッションが無効であることを前提としています。 細密なパーミッションが有効な場合は、操作が異なることがあります。

参照：
- 第 8 章、「細密なパーミッション」 (253 ページ)
- 第 6 章、「ユーザパーミッションの管理」 (167 ページ)

システム定義の役割の概要

システム標準の役割、値、およびタスクについて説明します。

次に、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。

役割	*role_granted* の値	説明
システム管理者	**sa_role**	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**	セキュリティ関連作業の実行
オペレータ	**oper_role**	データベースサーバ全体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

役割	*role_granted* の値	説明
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
ナビゲーションサーバ	**navigator_role**	navigator_role は syssrvroles に存在しているが、この役割は SAP ASE では使用されておらず、廃止された

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。

sa_serverprivs_role はシステム管理者に対して自動的に有効化されるユーザ定義の役割です。

参照：
- 細密なパーミッションによる sa_serverprivs_role の追加 (267 ページ)

システム標準の役割

システム定義の役割は、システムアクセスを認可されたユーザに制限します。

システム管理者の権限
システム管理者は、アプリケーション固有でないタスクを処理し、SAP ASE の任意アクセス制御システムの外部で作業します。

システム管理者の役割は、通常は特定の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、SAP ASE 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、システム管理者は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。 たとえば、**disk init**、**disk refit**、**disk reinit**、**shutdown**、**kill**、**disk mirror**、**mount**、**unmount**、いくつかのモニタリングコマンドなどです。

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、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、セキュリティに関係する次のような作業を実行します。

- 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の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の付与
- 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)。
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、データベース所有者がログインできるように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、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、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。

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
高可用性サブシステムを設定してコマンドとストアドプロシージャでプライマリサーバとコンパニオンサーバを管理するには、高可用性の役割が必要です。 『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、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** - あらかじめ定義されたジョブを使用できますが、ジョブを作成したり変更したり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
create service、**create existing service**、**drop service**、および **alter service** コマンドを実行するために、Web Services サブシステムで使用されます。『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

役割を実装する前に責任と位置を計画します。

ユーザ定義の役割を実装する前に、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します。 たとえば、役割名の末尾には “_role” を付けます。 SAP ASE 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に直接付与されるユーザ定義の役割の名前は、どのログインやログインプロファイルの名前とも重複していてはいけません。 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。

注意： sa_serverprivs_role はシステム管理者に対してログイン時に自動的に有効化されるユーザ定義の役割です。

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。

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。

『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。

役割のパスワードの追加または削除

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、役割をアクティブ化するときに、その役割を付与されたユーザが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、**alter role** コマンドを使用します。

構文は次のとおりです。

```
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。

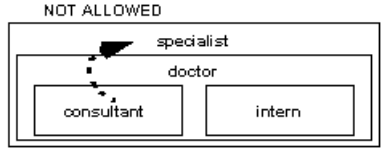
役割の階層を作成するときは、次の規則に従います。

- 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、明示的および暗黙的に付与された権限の図では、役割 ”specialist" に役割 ”doctor” がすでに含まれており、”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。
  明示的および暗黙的に付与された権限の図では、“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。

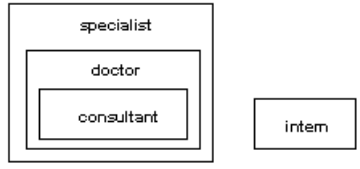
ただし、“intern” には、役割 “consultant” の権限はありません。これは、“intern” には役割 “consultant” が直接的にも間接的にも含まれないためです。



- 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、付与者に含まれる役割に対する役割の付与の図では、役割 “specialist” を役割 “consultant” に付与することはできません。“consultant” はすでに “specialist” に含まれているためです。



- 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他の役割が含まれる場合、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、メンバシップでの相互排他性の図で、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。



- 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、メンバシップでの相互排他性の図の階層で、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。

参照：
- 役割のアクティブ化と非アクティブ化 (100 ページ)

ログイン時のデフォルトアクティブ化

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 SAP ASE 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、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
```

役割アクティブ化の条件

システムセキュリティ担当者は、**grant role** 文で **where** 句を指定することによって、ユーザが特定の役割を持つことができる条件を制御できます。

SAP ASE は、ユーザが役割を設定するときまたはログイン時 (デフォルトの役割または自動的にアクティブ化される役割の場合) に、役割のアクティブ化述部と呼ばれる条件を評価します。 役割のアクティブ化述部が true と評価された場合にのみ、役割がアクティブ化されます。

参照：

- 述部付きの役割のアクティブ化 (240 ページ)

ユーザ定義の役割の削除

システムセキュリティ担当者は **drop role** コマンドを使用して役割を削除します。

構文は次のとおりです。

```
drop role role_name [with override]
```

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たアクセス権限がすべて取り消されます。

with override オプションを指定しない場合は、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、役割を削除する必要があります。 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、その役割におけるユーザの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

アクセス権限を得るには、役割をアクティブ化する必要があります。

デフォルトの役割は、ログイン時にアクティブ化されます。 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。 デフォルトの役割のアク

ティブ化述部がログイン時に false と評価された場合、役割は暗黙的に無視され、非アクティブのままになります。

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、**withpasswd** 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
```

アクティブ化述部を使用して役割が付与されている場合、アクティブ化述部が評価されます。 述部が true と評価されると、役割が有効になります。true と評価されなかった場合、役割は非アクティブのままになり、サーバはエラーメッセージを返します。

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なる点を考慮に入れておいてください。

役割に関する情報の表示

役割に関する情報を表示するには、次のシステムプロシージャと関数を使用します。

表示する情報	使用するシステムプロシージャ/関数
役割名の役割 ID	**role_id** システム関数
役割 ID の役割名	**role_name** システム関数
システム標準の役割	**show_role** システム関数
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
2 つの役割が相互排他的かどうか	**mut_excl_roles** システム関数
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
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

表示する情報	使用するシステムプロシージャ/関数
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
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

役割 ID と役割名の表示

役割 ID または役割名を表示するには、**role_id** 関数と **role_name** 関数を使用します。

役割名がわかっている場合に役割 ID を表示するには、**role_ID** 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。 役割が有効な場合、**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。

構文は次のとおりです。

```
show_role()
```

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。 データベース所有者が **setuser** を使用して別のユーザになり代わった後で **show_role** を実行した場合、**show_role** は、なり代わるユーザのアクティブなシステム標準の役割ではなく、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 **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したり、役割の階層ツリー全体をテーブル形式で表示したりすることができます。

構文は次のとおりです。

```
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。

構文は次のとおりです。

```
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** 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。

構文は次のとおりです。

```
mut_excl_roles(role1, role2, {membership | activation})
```

すべてのユーザが **mut_excl_roles** 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

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、**sp_activeroles** を使用します。

構文は次のとおりです。

```
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ようにするには、そのプロシージャの中で **has_role** を使用します。

has_role は、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ます。

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がストアドプロシージャを実行でき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** を使用して、呼び出しを行うユーザにプロシージャを実行するための適切な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 または 2 を返します。 所有していなければ、0 を返します。

たとえば、**has_role** を使用して、ユーザが役割 **sa_role** を持っているかどうかを確認するプロシージャは次のとおりです。

```
create proc test_proc
as
if (has_role("sa_role",0) = 0)
begin
print "You do not have the correct role"
return -1
end
else
print "You have System Administrator role"
return 0
```

役割の付与または取り消し

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。

作業	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**

作業	必要な役割	コマンド
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、役割、またはログインプロファイルに役割を付与するには、**grant role** コマンドを使用します。

構文は次のとおりです。

```
grant role role_name
    [where pred_expression]    to {username | rolename |
login_profile_name }
```

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、役割、ログインプロファイルに付与されます。 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
```

注意： 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。

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

参照：

- ログインプロファイルへの役割の付与 (61 ページ)

grant コマンドと役割について

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、直接的にまたはグループを通じて、指定した役割を*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 にシステムセキュリティ担当者の役割が付与され、**sso_role** に `sales` テーブルのパーミッションが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は個人に付与されているパーミッションより優先されるので、John は、**sso_role** をアクティブ化していれば `sales` にアクセスできます。

参照：
- 役割に関する情報の表示 (101 ページ)

役割の取り消し

ユーザ、他の役割、およびログインプロファイルから役割を取り消すには、**revoke role** を使用します。

構文は次のとおりです。

```
revoke role role_name [{, role_name}...]from grantee [{,
grantee}...]
```

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、役割、ログインプロファイルから取り消されます。

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 SAP ASE では、役割のパスワードは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。 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。

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、役割のパスワードが暗号化され、SHA256 ダイジェストとして格納されます。

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。ダウングレード時に、役割のパスワードがトランケートされ、役割はロックされます。 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の考慮事項

パスワードは、自動的に標準の形式、つまり標準化された汎用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。 15.7 よりも前のバージョンでは、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。

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。

ロックされた役割についての情報は、`syssrvroles` システムテーブルに格納されます。

- `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ときの `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ると、これらの値は 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ない。

lockreason の値	locksuid の値	役割の lockreason の説明
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た。
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	役割のアクティブ化の失敗回数がログイン失敗の最大回数に達したため、役割がロックされた。

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。

役割パスワードに適用されるログインパスワードポリシーの確認

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

高可用性機能によって、プライマリサーバとセカンダリサーバ間でパスワードポリシーオプションが同期化されます。

サポートされているパスワードオプションは次のとおりです。

- **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**

SAP ASE 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すべて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 password   password 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のパスワードポリシーチェックの不整合を示します。

SAP ASE での役割の設定

役割機能を使用する前に、SAP ASE 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** コマンドの具体的な数

SAP ASE のアップグレード

アップグレードプロセス中、SAP ASE はカラム `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。

SAP ASE のダウングレード

SAP ASE をダウングレードすると、役割のパスワードはトランケートされ、ロックされます。

さらに、SAP ASE では、役割のパスワードについて **allow password downgrade** の使用がサポートされていません。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、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、SAP ASE 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。

SAP ASE 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、SAP ASE 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 SAP ASE 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 5 章　外部認証

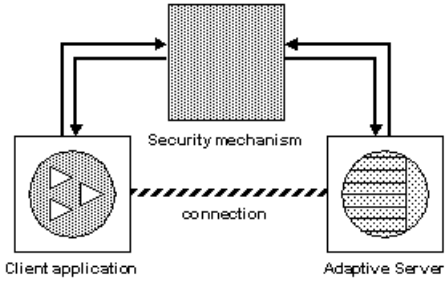
SAP ASE の外部のレポジトリに保管されている認証データを使用してユーザを認証できます。

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。 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。

SAP ASE でのネットワークベースのセキュリティの設定

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。



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、SAP ASE にクレデンシャルを送信します。

3. SAP ASE 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 SAP ASE 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、SAP ASE 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、SAP ASE にデータパケットを送信します。
3. SAP ASE 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. SAP ASE 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。

SAP ASE のセキュリティサービス

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。

次を使用できます。

- 統一化ログイン - ユーザを一度だけ認証する。ユーザは、SAP ASE 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あり、サーバは相互認証を要求できない。
- 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。
- 順序不整合のチェック - 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 - クライアントがサーバ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。 現時点では、CIS からのリモートサーバ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。

参照：

- 使用できるセキュリティサービスの情報の取得 (131 ページ)

ネットワークベースのセキュリティの管理

ネットワークベースのセキュリティの管理方法。

1. 設定ファイル (`libtcl.cfg`、`objectid.dat`、および `interfaces` (またはディレクトリサービス)) を編集します。

   使用しているプラットフォームの「セキュリティの設定ファイル」および『Open Client/Server 設定ガイド』を参照してください。
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 SAP ASE と Backup Server 用のログインが作成されていることを確認します。

   「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」およびセキュリティメカニズムのマニュアルを参照してください。
3. **sp_configure** を使用してインストールのセキュリティを設定します。
4. サーバを再起動して、**use security services** パラメータをアクティブ化します。
5. 企業全体のログインをサポートするため、**create login** を使用して、ログインアカウントを追加します。 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します。

   「統一化ログインをサポートするためのログインの追加」を参照してください。
6. **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します。

   「リモート接続での Kerberos セキュリティの確立」を参照してください。
7. **isql_r** や Open Client ClientLibrary を使い、使用するセキュリティサービスを指定して、SAP ASE に接続します。

   「サーバへの接続とセキュリティサービスの使用」を参照してください。『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」および使用しているプラットフォームの『Open Client/Server 設定ガイド』も参照してください。
8. **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します。 SAP ASE 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ます。

セキュリティの設定ファイル

設定ファイルは、インストール時に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。

次に設定ファイルの名前とロケーションを示します。

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。	*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

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。

SAP ASE 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、新しいサーバをネットワークに追加するときやサーバを新しいアドレスに移動するときに複数の interfaces ファイルを更新する必要がない点です。 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のセキュリティを使用するための libtcl.cfg の準備

ネットワークベースのセキュリティのために libtcl.cfg または libtcl64.cfg を編集します。

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
```

構文の説明は次のとおりです。

- *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
```

構文の説明は次のとおりです。

- *provider* には、セキュリティメカニズムのローカル名が入ります。 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` で定義されます。
  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

  デフォルトではないローカルメカニズム名を使用する場合は、`objectid.dat` ファイルのローカル名を変更します。
- *driver* は、セキュリティドライバの名前です。 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` です。
- *init-string* は、ドライバの初期設定文字列です。 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ります。
  - Kerberos ドライバ - 次に *init-string* の構文を示します。ここで、*realm* は Kerberos のデフォルトのレルム名です。

    ```
    secbase=@realm
    ```
  - Windows NT LAN Manager - *init-string* は適用されません。

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。 SAP ASE をインストールした後で、すで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。

SAP ASE 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すでに含まれています。

- [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、グローバルオブジェクト識別子をローカル名にマップします。また、このファイルには、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
```

たとえば、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プします。

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。
- 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。 SAP ASE サーバの名前を Windows NT LAN Manager にユーザとして定義し、SAP ASE 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。

SAP ASE 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、次の両方の条件が満たされていれば、SAP ASE 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。

ユーザに対する認証の要求

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要求するには、**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

`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。

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 SAP ASE 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、SAP ASE 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 SAP ASE 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、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 SAP ASE 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。

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

ログイン名は、有効な識別子でなければなりません。

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。

次の表に、ログイン名に使用されている無効な文字がどのように変換されるかを示します。

無効文字	変換後の文字
• 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 _
• 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 #

『リファレンスマニュアル』の「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。

暗号化によるメッセージの機密保持の要求

SAP ASE 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

SAP ASE では、**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、サーバ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、サーバに必要なメモリが、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「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

有効なログインとして定義されているユーザだけに SAP ASE の使用を許可するのか、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のかを検討します。

認証済みのクレデンシャルを使用してユーザがログインすると、SAP ASE は次の処理を行います。

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、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。

デフォルトログインを定義するには、デフォルトログインを `master..syslogins` に追加し、**sp_configure** を使用します。

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、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ます。

サーバにログインを追加し、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するには、次の一般的な手順に従います。

1. システムセキュリティ担当者が **create login** を使用して、ユーザのログインを追加します。
   「ログインアカウントの作成」を参照してください。
2. システム管理者またはデータベース所有者が **sp_adduser** を使用して、ユーザを 1 つ以上のデータベースに追加します。
   「データベースへのユーザの追加」を参照してください。
3. システム管理者またはデータベース所有者が **sp_changegroup** を使用して、ユーザをデータベースのグループに追加します。
   「グループメンバシップの変更」を参照してください。
4. システム管理者またはシステムセキュリティ担当者が **grant role** を使用して、ユーザにシステムの役割を付与します。
   「役割の付与または取り消し」を参照してください。
5. システムセキュリティ担当者が **create role** と **grant role** を使用して、ユーザ定義の役割を作成し、その役割をユーザに付与します。

「役割の付与」を参照してください。

6. データベースオブジェクト所有者が、データベースオブジェクトへのアクセス権を付与します。
   「ユーザパーミッションの管理」を参照してください。

参照：

- ログインアカウントの作成 (14 ページ)
- データベースへのユーザの追加 (65 ページ)
- グループメンバシップの変更 (70 ページ)
- 役割の付与または取り消し (104 ページ)
- 役割の付与 (105 ページ)
- 第 6 章、「ユーザパーミッションの管理」 (167 ページ)

リモート接続での Kerberos セキュリティの確立

SAP ASE 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 SAP ASE 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。

SAP ASE 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。

SAP ASE 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 を介したリモートサーバへの一般的な分散クエリ処理要求で、SAP ASE のリモートプロシージャコールを要求できます。 リモートサーバ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「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、両方のサーバで統一化ログインを使用できます。

次の 2 つの方法のうちのどちらかを使用します。

- 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" と定義します。 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 を実行します。 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はありません。
- 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しくみとして、Open Client Client-Library/C に **ct_remote_pwd** ルーチンが用意されています。 『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
```

セキュリティサービスによるサーバへの接続

isql ユーティリティと **bcp** ユーティリティでは、接続に対して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。

isql ユーティリティと **bcp** ユーティリティには、次のコマンドラインオプションが含まれています。

- `-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` - セキュリティメカニズムに定義された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*-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 の後にキー文字オプションの *security_options* 文字列を続けると、他のセキュリティサービスを有効にできます。 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** - 接続を確立するための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*-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。

ClientLibrary を使用して SAP ASE 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。

クライアントのセキュリティメカニズム

SAP ASE 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 SAP ASE 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)、SAP ASE 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

どのセキュリティメカニズムおよびセキュリティサービスがサポートされているのか、現在のセッションに対してどのセキュリティサービスがアクティブなのか、特定のセキュリティサービスが現在のセッションに対して有効になっているかどうかなどを調べることができます。

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
システムテーブル `syssecmechs` には、サポートされ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。

サポートされ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が有効かどうかを調べるには、**is_sec_service_on** 関数を使用します。ここで、*security_service_nm* は使用可能な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

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。

SAP ASE では、Kerberos は次のようにサポートされます。

- CyberSafe Kerberos ライブラリ、バージョン 1.3.2.1
- MIT Kerberos ライブラリ、バージョン 1.4.3 (Windows 64 ビット版の場合は バージョン 4.0.1)
- 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

SAP ASE における Kerberos の相互運用性です。

この表は、各種 Kerberos がサポートされるプラットフォームを示しています。

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
HP-UX Itanium 64 ビット版	MIT	MIT Kerberos 1.4.1

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
HP-UX PA Risc 64 ビット版	MIT	MIT Kerberos 1.4.3
IBM AIX 64 ビット版	MIT	Cybersafe TrustBroker 2.1、MIT Kerberos 1.4.3
Linux on Power 64 ビット版	MIT	MIT Kerberos 1.4.1
Linux Opteron 64 ビット版	MIT	MIT Kerberos 1.4.3、ネイティブ
Solaris 64 ビット版	MIT	Cybersafe TrustBroker 2.1、MIT Kerberos 1.4.1、ネイティブ
Solaris Opteron 64 ビット版	MIT、CSF	MIT Kerberos 1.4.2
Windows Opteron X64	AD	Cybersafe TrustBroker 2.1、MIT Kerberos 4.0.1

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 1.4.3 (Windows 64 ビット版の場合はバージョン 4.0.1)

SAP ASE での CyberSafe バージョン互換性の詳細については、CyberSafe の Web サイトを参照してください。 https://cybersafe.com/content/sybase-product-compatability.

Kerberos 環境での SAP ASE サーバの起動

Kerberos 環境で SAP ASE サーバを起動するには、サーバ名を 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 SAP ASE サーバ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 SAP ASE 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 2.1
      - MIT Kerberos バージョン 1.4.3 (Windows 64 ビット版の場合は バージョン 4.0.1)
   b) Kerberos KDC サーバを別の専用マシンにインストールします。

      CyberSafe TrustBroker バージョン 2.1、MIT Kerberos バージョン.1.8.2、Microsoft Windows Active Directory の KDC は、SAP ASE 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。

   この後の手順は Kerberos クライアントマシンで実行します。
2. SAP ASE 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 SAP ASE 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。

SAP ASE 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。 SAP ASE 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、SAP ASE 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. SAP ASE サーバを起動し、**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 SAP ASE 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。
   a) 設定パラメータ **use security services** を 1 に変更します。
   ```
   sp_configure 'use security services', 1
   ```
   b) 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. SAP ASE サーバを停止し、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、`libtcl.cfg` ファイルまたは `libtcl64.cfg` ファイルは `$SYBASE/$SYBASE_OCS/config/` にあります。 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。 次の例では、“csfkrb5” で始まる行はそれぞれ 1 つの行ですが、スペースの関係で分割されています。

  ```
  [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. サーバを再起動します。 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

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。

複数のインスタンスを実行中の場合は、サーバ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。

サーバのプリンシパル名

サーバ名を指定するには、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 SAP ASE サーバの名前です。 別の名前を指定するには、SAP ASE 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>
```

プリンシパル名を設定すると、この変数の値を使用して Kerberos に対する自己認証が行われます。

サーバの起動時に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>
```

Kerberos セキュリティメカニズムを有効にしてサーバを起動すると、`-k` オプションで指定されているプリンシパル名が Kerberos 認証に使用されます。 `-k` オプションが指定されていない場合、SYBASE_PRINCIPAL 環境変数で指定されているプリンシパル名が使用されます。 どちらも指定されていない場合は、認証にサーバ名が使用されます。

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。 次に例を示します。

  ```
  export SYBASE_PRINCIPAL=""
  ```

例

この例のサーバ名は「secure_rs」で、レルム名は「MYREALM.COM」です。 コマンドラインでは、`-s` パラメータを指定した **dataserver** でサーバ名が指定されます。 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
```

デフォルトのプリンシパル名は、“secure_ase@MYREALM.COM” です。 `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、サーバの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。

ユーザプリンシパル名と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)
```

構文の説明は次のとおりです。

- **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、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、SAP ASE 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 SAP ASE を設定します。「Kerberos の使用」と Open Client/Server のマニュアルを参照してください。 また、Sybase Web サイトにあるホワイトペーパー「Configuring Kerberos for Sybase」も参照してください。
   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」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込みパーミッションと実行パーミッションが必要です。

注意： "sybase" ユーザにのみ読み込み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。

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、SAP ASE サーバ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 認証を使用した SAP ASE へのログインの確認

Kerberos 認証を使用して SAP ASE 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 SAP ASE 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

sybmapname は、SAP ASE 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、**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) として構築した後に、適切なパス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 SAP ASE 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 SAP ASE 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>
```

Kerberos による同時認証

SAP ASE は、Kerberos による同時認証をサポートしています。

Kerberos 認証を使用した同時ログインがある場合は、SAP ASE によって複数の Kerberos 認証セッションが確立されます。

SAP ASE 15.0.3 より前の バージョンでは、Kerberos 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。

LDAP ユーザ認証のための SAP ASE 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 SAP ASE 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、SAP ASE 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、SAP ASE バージョン 12.5.1 以降で使用可能)
- 検索 DN (SAP ASE 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。SAP ASE 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
```

構文の説明は次のとおりです。

- *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。 filter 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 がリスナポートに接続します。
2. リスナが接続を受け入れます。
3. Open Client が内部ログインレコードを送信します。
4. SAP ASE サーバがログインレコードを読み込みます。
5. SAP ASE が、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、SAP ASE は LDAP サーバに検索要求を送信します。
8. LDAP サーバが検索結果を返します。
9. 検索結果に基づいて、ログインが受け入れられるか拒否されます。

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 が SAP ASE のリスナポートに接続します。
2. SAP ASE リスナが接続を受け付けます。
3. Open Client が内部ログインレコードを送信します。
4. SAP ASE がログインレコードを読み込みます。
5. SAP ASE が、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 SAP ASE 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 サーバが、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. SAP ASE が、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 サーバが検索結果を返します。
9. SAP ASE が、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. SAP ASE が、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 サーバが、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、SAP ASE は LDAP サーバに検索要求を送信します。
13. LDAP サーバが検索結果を返します。
14. SAP ASE が、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、SAP ASE 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。

新しい SAP ASE での LDAP の設定

SAP ASE を LDAP 認証向けに設定することができます。

1. 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 SAP ASE サーバの 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 SAP ASE を LDAP にマイグレートします。

1. SAP ASE 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。

生成 DN 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。
- 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。 SAP ASE は、このフィルタのワイルドカードを認証対象となる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
  ```
- 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。
  1. 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'
```

  2. プライマリ URL を設定します。

```
sp_ldapadmin set_primary_url, 'ldap://hostname:389/
```

  3. 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、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、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))'
```

  SAP ASE 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、SAP ASE 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

SAP ASE が LDAP サーバから取得してクライアントに渡す、LDAP ユーザ認証関連の情報メッセージが 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 SAP ASE サーバにログインした場合は、次のメッセージが表示されます。

```
Your password will expire in <number> days.
```

- 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 SAP ASE にログインすると、次のメッセージ 4002 が表示されます。

```
Login failed
```

  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ます。

```
Your LDAP password has expired.
```

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、SAP ASE 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、SAP ASE 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。

SAP ASE 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、SAP ASE サーバはセカンダリ 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、セカンダリ URL にフェールオーバしません。

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 SAP ASE 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、SAP ASE 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、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。

「LDAP による syslogins の変更」の表に、SAP ASE でログイン時に行われる `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。

LDAP による syslogins の変更は次のとおりです。

syslogins にそのユーザのローがすでに存在する	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

LDAP サーバによって認証された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モニタリングされます。

- INITIAL - LDAP ユーザ認証が設定されていないことを示す。
- RESET - 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。 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： SAP ASE サーバがセカンダリ 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。

SAP ASE 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。

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]
- 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

LDAP サーバのステータスの移行を確認するには、**sp_ldapadmin** コマンドを使用します。

表 6 : 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

初期状態	最終状態
SUSPENDED	RESET

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 7 : 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 8 : 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、SAP ASE 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 9 : SAP ASE 再起動時のステータスの移行

初期状態	最終状態
INITIAL	INITIAL

初期状態	最終状態
RESET	RESET
READY	READY
ACTIVE	READY
FAILED	FAILED
SUSPENDED	SUSPENDED

SAP ASE 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 10 : 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 11 : 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

SAP ASE 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 SAP ASE/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 SAP ASE/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、着信接続数が **max_ldapua_native_threads** に達した場合の SAP ASE 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、SAP ASE 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、SAP ASE 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** - 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。

SAP ASE 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すで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 1

LDAP ユーザを "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、数百台のサーバ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" がサーバ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サーバで "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、サーバ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、SAP ASE 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 SAP ASE 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
```

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サーバ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、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が更新されます。

次に例を示します。

1. ユーザには、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。SAP ASE 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` を使用してサーバにログインすると、`user_ase` のパスワードは `user_ldappassword` に更新されます。

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プされていると、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、複数の外部ユーザを単一のユーザにマップすることはできません。

LDAP ユーザ認証エラーのトラブルシューティング

SAP ASE 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、SAP ASE 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、SAP ASE と LDAP サーバ間でのデータ転送の安全性を確保できます。

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、セキュア接続が次のように設定されます。ここで、*servername* は現在のサーバの名前です。

   - `$SYBASE_CERTDIR` を定義した場合は、`$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。
   - `$SYBASE_CERTDIR` を定義しなかった場合は、`$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. サーバ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** を使用して、`ldaps://` URL を `ldap://` URL の代わりに指定して、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、プレーンテキストでの 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

reinit_descriptors コマンドは、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、CA (認証局) によって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。
  SAP ASE 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。

自動的な LDAP ユーザ認証とフェールバック

SAP ASE では、セカンダリ LDAP サーバがサポートされています。

sp_ldapadminset_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

LDAP サーバを自動的にアクティブにするには、**'set_failback_interval'** を使用します。

構文は次のとおりです。

```
sp_ldapadmin 'set_failback_interval', time_in_minutes
```

構文の説明は次のとおりです。

time_in_minutes は、-1 ～ 1440 分 (24 時間) の値です。

- 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)
```

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'
```

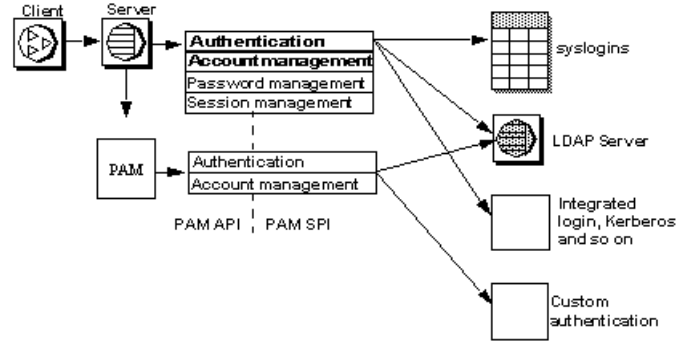
```
The LDAP property 'set_failback_interval' is set to '15
minutes'.
```

PAM を使用する認証のための SAP ASE 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 SAP ASE 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。



SAP ASE 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。

SAP ASE での PCI 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。 SAP ASE 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 サポートが有効になるようにオペレーティングシステムを設定します。

- 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 required /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
```

```
aseauthrequired /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 SAP ASE 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。

構文は次のとおりです。

```
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。 これはデフォルト値です。
- 1 - SAP ASE 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。

ログインと PAM ユーザアカウント

enable PAM user authentication を設定し、SAP ASE 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 SAP ASE 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、SAP ASE 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、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。

次の表に、ログイン時の syslogins テーブルの変更を示します。 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。

syslogins にそのユーザのローがすでに存在する	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

syslogins にそのユーザのローがすでに存在する	PAM 認証に成功	syslogins の変更
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 SAP ASE を設定します。

また、機能拡張された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 SAP ASE を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** 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。

alter login と **create login** で次のパラメータを使用します。

- **ASE** - `syslogins` テーブルに格納されているパスワードを使用する SAP ASE 内部認証を使用します。
- **LDAP** - LDAP サーバによる外部認証を使用します。
- **PAM** - PAM による外部認証を使用します。
- **ANY** - デフォルトのユーザ認証メソッドです。 ユーザに対して ANY 認証を指定すると、SAP ASE 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、SAP ASE の認証が使用されます。

SAP ASE 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プ

次の構文で **sp_maplogin** を使用してログインをマップできます。

構文は次のとおりです。

```
sp_maplogin (authentication_mech | null),
(client_username | null), (action | login_name | null)
```

次の例は、外部ユーザ "jsmith" をユーザ "guest" にマップします。 認証が行われると、"jsmith" は "guest" の権限を得ます。 監査ログインレコードには、*client_username* とユーザ名の両方が表示されます。

```
sp_maplogin NULL, "jsmith", "guest"
```

次の例では、LDAP で認証されたすべての外部ユーザのログインが存在しない場合に、ログインが新しく作成されます。

```
sp_maplogin LDAP, NULL, "create login"
```

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。

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。

構文は次のとおりです。

```
sp_helpmaplogin [ (authentication_mech | null), (client_username |
null) ]
```

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、SAP ASE 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

SAP ASE 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、SAP ASE 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、SAP ASE 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、Joe は SAP ASE 認証を使用して認証されます。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
```

SAP ASE 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 6 章　ユーザパーミッションの管理

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。

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。

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、次のことを行うため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。

細密なパーミッションが有効な場合は、操作が異なることがあります。

参照：

- 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 (195 ページ)
- 第 8 章、「細密なパーミッション」 (253 ページ)

データベース作成用のパーミッションの付与

create database コマンドを使用するための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 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、すべての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。

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 を使用すると、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。 新しい所有者はサーバ上にすで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

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。

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、**create encryption key** および **create**

trigger 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)。

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。

- 次のコマンドの使用: **create default**、**create procedure**、**create rule**、**create table**、**create view**。
  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。
- システムテーブルのデフォルトパーミッション
- **dbcc** コマンドの使用: **checkalloc**、**checkcatalog**、**checkdb**、**checkindex**、**checkstorage**、**checktable**、**checkverify**、**fix_text**、**indexalloc**、**reindex**、**tablealloc**、**textalloc**、**tune**。

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。

データベースオブジェクト所有者の権限

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** コマンドを使用し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。

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または取り消し

grant コマンドと **revoke** コマンドを使用してパーミッションを制御します。

grant コマンドと **revoke** コマンドを使用して制御されるパーミッションは次のとおりです。

- 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、パーミッションが明示的に付与されている必要があります。 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 12 : 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。 『リファレンスマニュアル: コマンド』を参照してください。

where 句を使用して `select`、`update`、および `delete` の各パーミッションを付与することにより、**where** 句の条件に基づいて、ローに対するアクセスをロー単位で制限することができます。

具体的 ID

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ます。

具体的 ID によって、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。

注意： SAP ASE 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場合、そのプロシージャを実行するための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 select 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 `id` カラムと `descr` カラムを参照できます。

```
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** をオンにした場合は、**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。

以下は、update 文と delete 文を実行するための ANSI パーミッションをまとめたものです。

	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** 句にあ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** 句にあるすべてのカラムに対する **select** パーミッション

ansi_permissions がオンの場合に、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

オブジェクトアクセスパーミッションの付与の例を次に示します。

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
```

次の文は、12 月の給与を更新するためのパーミッションを給与担当者に付与します。

```
grant update (salary)
on employee
where date_part(month, getdate()) = 12
to payroll_role
```

オブジェクトアクセスパーミッションの取り消しの例

オブジェクトアクセスパーミッションの取り消しの例を次に示します。

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。

『リファレンスマニュアル: コマンド』で **dbcc** 構文の詳細を参照してください。

サーバワイドな dbcc コマンドとデータベース固有の dbcc コマンド

dbcc コマンドには、データベース固有のものとサーバワイドなものがあります。

- データベース固有の **dbcc** コマンドは、特定のターゲットデータベースに対して実行されます (**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。
- **tune** などのサーバワイドな **dbcc** コマンドは、サーバ全体に作用しますが、特定のデータベースには関連付けられません。 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。

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。

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 **grantdbccheckstorage** 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 **grantdbccheckstorage** 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な **dbcc** コマンド
- **storage_admin_role** に対して付与される **grantdbccheckstorage** などの、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

システムテーブルのパーミッション

システムテーブルを使用するためのパーミッションは、他のテーブルのパーミッションと同様に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 SAP ASE 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！ SAP ASE 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、SAP ASE 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	sysresourceli-mits	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。
- **sso_role** に sysdatabases の **select** を付与する。
- 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
```

その後で、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。競合が発生した場合は、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

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。

注意： **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。

```
SQL statement on line number 1 contains Non-ANSI text. The error is
caused due to the use of DBCC.
```

別のユーザのパーミッションの取得

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は 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。
- 「*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。

setuser コマンド

データベース所有者は、**setuser** を使用して、別のユーザになり代わることができます。

次の場合に **setuser** を使用します。

- 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、このコマンドはすで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。 SAP ASE 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

SAP ASE 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 SAP ASE 内の別のログインになり代わることができます。

警告！ 他のユーザの 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]
```

構文の説明は次のとおりです。

- *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。

たとえば、次の例では **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” がすで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:Set session authorization permission denied
because the target login has a role that you do not have and you have
been restricted from using.
```

“joe” 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** を 1 つのトランザクション内で実行することはできません。
- 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
```

```
-------------------------------------------------
sallynsally
```

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
```

```
-------------------------------------------------
ralphguest
```

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
```

```
-------------------------------------------------
rudolphguest
```

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。

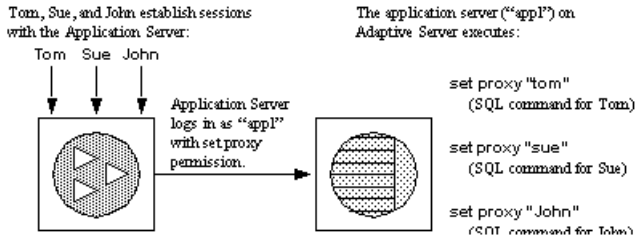
```
set proxy "ralph"
go
set proxy "sa"
go
select suser_name(), user_name()
go
```

```
---------------------------------------------
sadbo
```

アプリケーションによる代理権限の使用方法

代理権限を使用すると、アプリケーションサーバは汎用ログインを使用して SAP ASE サーバにログインし、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できます。

次の図は、汎用ログイン “appl” を使用して SAP ASE サーバにログインし、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サーバを示しています。 “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。



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

オブジェクトの所有権は、特定の所有者から別の所有者に譲渡できます。

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

注意： 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

所有権を譲渡する権限

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。

- データベース所有者が所有権を譲渡する場合は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限定して行われることもあれば、複数のオブジェクトを 1 つのコマンドで譲渡することもあります。 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

SAP が提供する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します。

SAP 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`
- `tempdb`

SAP が提供および管理する次のシステムオブジェクトの所有権は変更しないでください。

- `spt_` プレフィクスの付いたユーザテーブル
- **sp_** プレフィクスの付いたシステムストアドプロシージャ

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** パラメータを使用して所有権を譲渡できます。 ***** を使用し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 コマンド

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 がテーブル `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` 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` に対する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 delete、insert、eferences、select、update の各パーミッションがあります。 また、bill には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じた decrypt パーミッションもあります。 システムセキュリティ担当者が **preserve permissions** を使用して `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こ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： decrypt 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 alicep 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。

暗号化キーの所有者を変更しても、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。

パーミッションのレポート

システムプロシージャやシステムテーブルを使用して、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します。

レポートする情報	使用するもの
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_helprotect**

sysprotects テーブルに対する代理権限に関するクエリの実行

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` テーブルに対してクエリを実行します。

`action` カラムで、パーミッションを指定します。 たとえば、**set proxy** と **set session authorization** の `action` の値はどちらも 167 です。

次のクエリを実行します。

```
select * from sysprotects where action = 167
```

クエリ実行結果として、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

sp_who は、現在のすべての SAP ASE 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 `orignam`e 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、`orignam`e 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 システムテーブルに対してクエリを実行します。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のレポート

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。

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    predicate   gra
ntable
------- ------ -----   ------   ------     ------    -------    ----
------
dbo     judy   Grant   Select   titles     All       NULL       FALSE
dbo     judy   Grant   Update   titles     advance   NULL       FALSE
dbo     judy   Grant   Update   titles     notes     NULL       FALSE
dbo     judy   Grant   Update   titles     price     NULL       FALSE
dbo     judy   Grant   Update   titles     pub_id    NULL       FALSE
dbo     judy   Grant   Update   titles     pubdate   NULL       FALSE
dbo     judy   Grant   Update   titles     title     NULL       FALSE
dbo     judy   Grant   Update   titles     title_id  NULL       FALSE
dbo     judy   Grant   Update   titles     total_sales NULL     FALSE
dbo     judy   Grant   Update   titles     type      NULL       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、Judy は他のユーザに **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   predicate    gra
ntable
------- ------ -----   ------   ------     ------   -------   -------
dbo     judy    Grant   Select  publishers   all       NULL
       TRUE
```

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。

特定のテーブルに対するパーミッションのレポート

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_helprotect** を使用します。

sales テーブルに対す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_tables sales
```

```
grantor grantee type       action        object  column  predicate
 grantable------- ------
-----   ------   --------      ------   -------     -------
dbo    guest  Grant    Delete          sales   All    NULL     FALSE
dbo     guest  Grant    Delete
Statistics sales   All     NULL      FALSE
dbo    guest  Grant    Insert          sales   All    NULL     FALSE
dbo    guest  Grant    References      sales   All    NULL     FALSE
dbo     guest  Grant    Transfer
Table    sales   All     NULL      FALSE
dbo     guest  Grant    Transfer
Table    sales   All     NULL      FALSE
dbo    guest  Grant    Update          sales   All    NULL     FALSE
dbo     guest  Grant    Update
Statistics sales   All     NULL      FALSE
dbo    public Grant    Select          sales   All    NULL     FALSE
(1 row affected)
(return status = 0)
```

sp_helprotect 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

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、 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 SAP ASE 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

ビューを使用して表示できるデータに対してだけ、クエリや変更ができます。 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。

一部のユーザを、titles テーブル内の金銭と売上に関するカラムにアクセスできないようにする場合は、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。

```
permission denied
```

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。

参照：

- 所有権の連鎖の理解 (199 ページ)

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。

execute as owner または **execute as caller** を使用して、それぞれ所有者または呼び出し元に代わってランタイムパーミッションを検査し、DDL を実行して、オブジェクト名を解決するプロシージャを作成できます。

execute as owner または **execute as caller** を使用して定義されたプロシージャ、あるいは **executeas** 句なしで定義されたプロシージャを、**execute as owner** または **execute as caller** を使用して定義されたプロシージャ内にネストすることができま

す。また、**execute as owner** または **execute as caller** を使用して定義されたプロシージャを、**execute as** 句なしで定義されたプロシージャ内にネストすることもできます。

execute as owner を使用して作成されたプロシージャ内で **set session authorization** 文を使用することはできません。**execute as owner** として定義されていない、ネストされたプロシージャ内にこの文がある場合も同様です。

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

役割とストアドプロシージャ

grant execute コマンドを使用すると、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。

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、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は次のとおりです。

```
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
```

『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」で **has_role** の詳細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。

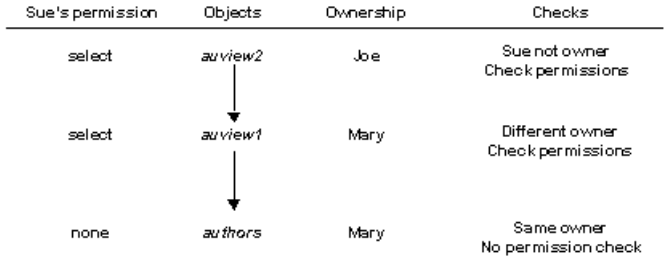
ただし、別のユーザが所有しているオブジェクトに従属するビューまたはストアドプロシージャを作成する場合は、自分が付与するパーミッションが、それらの他の所有者によって許可されているパーミッションに従属することに注意してください。

ビューと所有権の連鎖の例

所有権の連鎖とパーミッション検査の例を次に示します。

Joe が作成する auview2 というビューが Mary のビュー auview1 に従属する場合を例にとって、所有権の連鎖を見ていきます。 Joe は auview2 に対する **select** パーミッションを Sue に付与します。

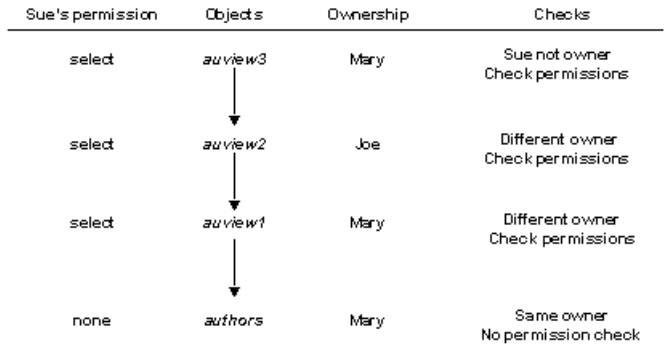
所有権の連鎖は次のようになります。



SAP ASE 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。



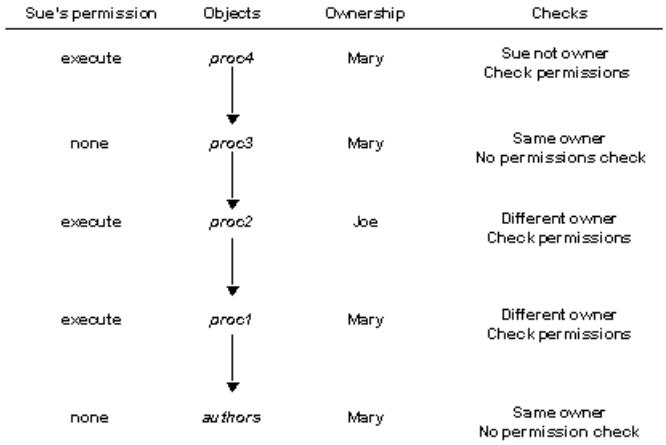
Sue が auview3 にアクセスすると、SAP ASE は、auview3、auview2、auview1 に対するパーミッションを検査します。 auview2 に対するパーミッションが Joe から Sue に付与され、auview3 と auview1 に対するパーミッション

が Mary から付与されていれば、SAP ASE はアクセスを許可します。 SAP ASE 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

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の例を次に示します。

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。



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。

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 のパーミッションが検査されます。 SAP ASE 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 SAP ASE 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。

「トリガ: 参照整合性の保持」(『Transact-SQL ユーザーズガイド』) を参照してください。

execute as owner または execute as caller を含むプロシージャの実行

execute as owner または **execute as caller** を使用して、それぞれ所有者または呼び出し元に代わってランタイムパーミッションを検査し、DDL を実行して、オブジェクト名を解決するプロシージャを作成できます。

execute as caller を使用してプロシージャを作成した場合、プロシージャ呼び出し元としてこれらの操作が実行されます。 **execute as owner** を使用してプロシージャを作成した場合は、プロシージャ所有者に代わってこれらの操作が実行されます。

execute as 句を省略した場合の動作は、SAP ASE 15.7 ESD #2 より前のバージョンと同じです。

プロシージャ内のすべての操作をプロシージャ所有者の権限に照らして検査する必要があるアプリケーションの場合、プロシージャ所有者として実行するプロシージャを作成すると便利です。 execute as owner を使用して作成されたプロシージャには、所有権の連鎖による暗黙的な付与の検査は適用されません。 アプリケーションのエンドユーザには、ストアドプロシージャに対する **execute** パーミッ

ションを除き、データベースにおける権限は必要ありません。 さらに、プロシージャによって実行される DDL はプロシージャ所有者に代わって実行され、プロシージャで作成されるオブジェクトはプロシージャ所有者が所有します。 そのため、管理者は、DDL コマンドに対する権限をアプリケーションユーザに付与する必要がありません。

個々のユーザに代わってパーミッションを検査する必要がある場合は、セッションユーザまたは呼び出し元として実行するプロシージャを作成する必要があります。 たとえば、プロシージャがアクセスするテーブルに、述部付きの権限による詳細なアクセス制御が適用される場合 (1 人のユーザに 1 つのローセットを表示する権利を付与し、別のユーザには別のローセットを表示する権利を付与する場合など)、**execute as caller** を使用します。 execute as caller を使用して作成されたプロシージャには、所有権の連鎖による暗黙的な付与の検査は適用されません。プロシージャで参照されるテーブルに述部付きの権限がある場合、それらの権限が適用されます。

execute as 句を省略した場合の動作は次のとおりです。

- プロシージャ所有者に代わってオブジェクト名が解決されます。
- プロシージャ呼び出し元に代わって DDL コマンドとデータベース間アクセスが実行されます。
- 呼び出し元に代わって、DML、**execute**、**transfer table**、**truncate table**、**delete statistics**、および **update statistics** のパーミッション検査が行われます。ただし、参照されるオブジェクトとプロシージャ間に所有権の連鎖が存在し、パーミッション検査が省略される場合を除きます。

execute as owner 句を指定した場合、プロシージャの動作は、実行の開始時における所有者に対する暗黙的な **set session authorization** の後に想定される動作に従います。 この動作は次のとおりです。

- プロシージャ所有者に代わってオブジェクト名が解決されます。 名前を所有者名で修飾せずにテーブルまたはその他のオブジェクトがプロシージャで参照されている場合、SAP ASE は、プロシージャ所有者が所有するその名前のテーブルを検索します。 そのようなテーブルがなければ、データベース所有者が所有するその名前のテーブルを検索します。
- プロシージャ所有者に代わって DDL コマンドとデータベース間アクセスが実行されます。
- 所有権の連鎖によるパーミッションの暗黙的な付与は行われません。
- アクセス制御チェックはすべて、プロシージャ所有者のパーミッション、グループ、システム役割、デフォルトのユーザ定義の役割、およびプロシージャの本文でアクティブ化された、所有者に付与されている役割に基づきます。 所有者のログインプロファイルに付与されている役割も考慮されます。 プロ

シージャの実行時にプロシージャ所有者がアクティブ化した役割のみが考慮されます。

- **execute as owner** または **execute as caller** を使用して定義されたプロシージャ、あるいは execute as 句なしで定義されたプロシージャを、**execute as owner** を使用して定義されたプロシージャ内にネストすることができます。 また、**execute as owner** を使用して定義されたプロシージャを、**execute as** 句なしで定義されたプロシージャ内にネストすることもできます。
  execute as owner プロシージャから呼び出されたプロシージャは、ネストされたプロシージャが **execute as owner** として定義されている場合を除き、呼び出し元のプロシージャの所有者として実行されます。
- プロシージャ内の動的 SQL 文は、**sp_procxmode** の **'Dynamic Ownership Chain'** の設定に関係なく、プロシージャ所有者のパーミッションで実行されます。
- テンポラリテーブルはセッションが所有するため、呼び出し元によってプロシージャの外部で作成されたテンポラリテーブルを、プロシージャ所有者がプロシージャ内で使用することができます。 この動作は、**set session authorization** コマンドがセッションで実行された後、テンポラリテーブルが使用可能であることを示しています。
- プロシージャ内で実行された文の監査レコードには、プロシージャ所有者の名前と **execute as owner** オプションが表示されます。
- **execute as owner** を使用して作成されたプロシージャ内で **set session authorization** 文を使用することはできません。**execute as owner** として定義されていない、ネストされたプロシージャ内にこの文がある場合も同様です。

execute as caller 句を指定した場合の動作は次のとおりです。

- 呼び出し元に代わってオブジェクトが解決されます。 名前を所有者名で修飾せずにテーブルまたはその他のオブジェクトがプロシージャで参照されている場合、SAP ASE は、プロシージャを呼び出したユーザが所有するその名前のテーブルを検索します。 そのようなテーブルがなければ、データベース所有者が所有するその名前のテーブルを検索します。
- 呼び出し元に代わって DDL コマンドとデータベース間アクセスが実行されます。
- 所有権の連鎖によるパーミッションの暗黙的な付与は行われません。
- 呼び出し元、呼び出し元のグループ、アクティブな役割およびシステム役割に代わって、パーミッションが検査されます。
- **execute as owner** または **execute as caller** を使用して定義されたプロシージャ、あるいは execute as 句なしで定義されたプロシージャを、**execute as caller** を使用して定義されたプロシージャ内にネストすることができます。 また、**execute as caller** を使用して定義されたプロシージャを、**execute as** 句なしで定義されたプロシージャ内にネストすることもできます。

execute as caller プロシージャから呼び出されたプロシージャは、ネストされたプロシージャが **execute as owner** として定義されている場合を除き、親プロシージャの呼び出し元に代わって実行されます。

- 動的 SQL は、**sp_procxmode** の **'Dynamic Ownership Chain'** の設定に関係なく、呼び出し元として実行されます。
- プロシージャの外部で作成されたテンポラリテーブルをプロシージャ内で使用することができます。
- プロシージャの各呼び出し元に従ってオブジェクトが解決されるため、プロシージャによるオブジェクト参照は `sysdepends` に入力されません。
- **select *** は `syscomments` で拡張されません。
- プロシージャ内のオブジェクトは、プロシージャを実行するユーザに解決する必要があるため、同じプロシージャに関するプロシージャキャッシュ内のプランはユーザ間で共有されません。 この理由により、多数のユーザがプロシージャを実行する場合、プロシージャキャッシュの使用量が増加することがあります。 特定のユーザがプロシージャを再度実行した場合は、そのユーザのプランが再利用されます。
- プロシージャ内で実行された文の監査レコードには、プロシージャ呼び出し元の名前と **execute as caller** 句が表示されます。

次の例では、ユーザ Jane が作成したプロシージャに **execute as** 句はありません。このプロシージャは、`jane.employee` から選択して `emp_interim` という名前の中間テーブルに挿入します。

```
create procedure p_emp
```

```
select * into emp_interim
 from jane.employee
```

```
grant execute on p_emp to bill
```

Bill がプロシージャを実行します。

```
exec jane.p_emp
```

- Jane が `p_emp` と `employee` を所有しているため、Bill には、`jane.employee` に対する select パーミッションは必要ありません。 `p_emp` に対する **execute** パーミッションを Bill に付与することによって、Jane は `employee` に対する **select** パーミッションを暗黙的に Bill に付与しています。
- Bill には、create table パーミッションが付与されている必要があります。`emp_interim` テーブルは、Bill が所有することになります。

次の例では、Jane が **execute as owner** 句を使用して同じ本文を持つプロシージャを作成し、Bill がプロシージャを実行します。

```
create procedure p_emp
```

```
with execute as owner as
```

```
    select * into emp_interim
    from jane.employee

grant execute on p_emp to bill
```

- Bill がプロシージャを適切に実行するには、**execute** パーミッションのみが必要です。
- emp_interim テーブルは、Jane に代わって作成されます。つまり、Jane が所有者です。 Jane に create table パーミッションがない場合、プロシージャは失敗します。

次の例では、Jane が **execute as caller** 句を使用して同じプロシージャを作成します。

```
create procedure p_emp

    with execute as caller as

    select * into emp_interim
    from jane.employee

grant execute on p_emp to bill
```

- Bill は、jane.employee に対する **select** パーミッションを持っている必要があります。 Jane が p_emp と employee の両方を所有していますが、暗黙的な付与の検査は行われません。 Bill に付与された述部付きの権限が jane.employee にある場合、述部がクエリに追加されます。
- Bill には、create table パーミッションが必要です。 emp_interim は、Bill に代わって作成されます。つまり、Bill が所有者です。

修飾されていない名前を使用してオブジェクトを参照するプロシージャの作成

execute as 句を使用しないプロシージャの作成の例と、**execute as owner** 句または **execute as caller** 句を使用したプロシージャの作成の例を次に示します。

次の例では、プロシージャに **execute as** 句はありません。

```
create procedure insert p

insert t1 (c1) values (100)

grant execute on insert p to bill
```

Bill がプロシージャを実行します。

```
exec jane.insert p
```

- Jane が所有する t1 という名前のテーブルを探すための検索が実行されます。 jane.t1 がなければ、dbo.t1 を探すための検索が実行されます。

- t1 が dbo.t1 に解決された場合、Bill は t1 に対する insert パーミッションを持っている必要があります。
- t1 が jane.t1 に解決された場合、jane.insert_p と jane.t1 間の所有権の連鎖により、暗黙的な insert パーミッションが Bill に付与されます。

次の例では、Jane が **execute as owner** を使用して上記と同じプロシージャを作成します。

```
create procedure insert p
with execute as owner as
    insert t1 (c1) values (100)
grant execute on insert p to bill
```

Bill がプロシージャを実行します。

```
exec jane.insert p
```

- Jane が所有する t1 という名前のテーブルを探すための検索が実行されます。jane.t1 がなければ、dbo.t1 を探すための検索が実行されます。
- t1 が dbo.t1 に解決された場合、t1 に対する insert パーミッションが Jane に付与されている必要があります。
- t1 が jane.t1 に解決された場合、このプロシージャは所有者として実行されているため、Jane はそのパーミッションを持っています。

次の例では、Jane が **execute as caller** を使用して上記と同じプロシージャを作成します。

```
create procedure insert p
    with execute as caller as
    insert t1 (c1) values (100)
grant execute on insert p to bill
```

Bill がプロシージャを実行します。

```
exec jane.insert p
```

- Bill が所有する t1 という名前のテーブルを探すための検索が実行されます。bill.t1 がなければ、dbo.t1 を探すための検索が実行されます。
- t1 が dbo.t1 に解決された場合、Bill は、t1 に対する insert パーミッションを持っている必要があります。

完全修飾名を使用して別のデータベースでネストされたプロシージャを呼び出すプロシージャ

execute as owner と **execute as caller** を使用したネストされたプロシージャの作成の例を次に示します。

次の例では、Jane が、完全修飾名を使用して別のデータベースでネストされたプロシージャを呼び出すプロシージャを作成します。 Jane に関連付けられたログインは、otherdb 内のユーザ Jane に解決されます。 この例では、**execute as owner** を使用します。

```
create procedure p master
with execute as owner
    as exec otherdb.jim.p_child
grant execute on p master to bill
```

Bill がプロシージャを実行します。

- SAP ASE は、otherdb 内のユーザ Jane が jim.p_child に対する **execute** パーミッションを持っているかどうかを検査します。
- execute as 句なしで jim.p_child が作成されている場合、p_child は Jane に代わって実行されます。
- **execute as owner** を使用して jim.p_child が作成されている場合、p_child は Jim に代わって実行されます。
- **execute as caller** を使用して jim.p_child が作成されている場合、p_child は Jane に代わって実行されます。

次の例では、Jane が **execute as caller** を使用して上記と同じプロシージャを作成します。 現在のデータベース内のユーザ Bill に関連付けられたログインは、otherdb 内のユーザ Bill に解決されます。

```
create procedure p master
with execute as caller
    as exec otherdb.jim.p_child
grant execute p master to bill
```

Bill がプロシージャを実行します。

```
exec jane.p_master
```

- SAP ASE は、otherdb 内の Bill が jim.p_child に対する **execute** パーミッションを持っているかどうかを検査します。

- execute as 句なしで jim.p_child が作成されている場合、p_child は Bill に代わって実行されます。
- **execute as owner** を使用して jim.p_child が作成されている場合、p_child は Jim に代わって実行されます。
- **execute as caller** を使用して jim.p_child が作成されている場合、p_child は Bill に代わって実行されます。

ローレベルアクセス制御

ローレベルアクセス制御によって、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。

ローレベルアクセス制御には次の機能があります。

- 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。

SAP ASE 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、使用され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、サーバを再起動する必要はありません。

grant 文で **where** 句を使用して、ローレベルのアクセス権を付与することもできます。 プライバシーポリシーが、他のテーブル内のデータや、サブクエリによって SQL の式全体に依存する場合は、この方法のローレベルアクセス制御を使用します。

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** 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**、**d**elete 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

テーブルを作成して、そのテーブルにアクセスルールをバインドすることができます。

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
```

SAP ASE は、テーブル 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"
```

次に例を示します。

```
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例

アクセスルールの作成の例を次に示します。

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)
```

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

(4 rows affected)

/* 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、SAP ASE 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。

次に例を示します。

```
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 で構成されます。 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。 SAP 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。

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によるパーミッションの設定

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、次の組み込み関数を使用します。

- **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。
- **get_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取得されます。
- **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。
- **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。

構文については、『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。

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。

- **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
  ```

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。 SAP ASE 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、**grant** 権限と **revoke** 権限を使用します。 `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制御チェックの中では、**select** 権限のみがチェックされます。

有効なユーザ
組み込み関数にはオブジェクト ID やホームデータベースがないため、データベース所有者は関数に対する `select` 権限を適切なユーザまたは役割に付与する必要があります。 SAP ASE は、ユーザのデフォルトデータベースをテンプレートとして使用して追加のデータベースに対する権限を特定します。 ただし、`select` 権限を付与できるのは、ユーザのデフォルトデータベースの所有者だけです。 1 つのデータベースで組み込み関数を使用することによってユーザが追加のデータベースを使用できないようにする場合は、追加のデータベースの所有者がそのデータベースのユーザまたは役割のパーミッションを明示的に取り消す必要があります。

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。 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**

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 13 : 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

属性	値
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

アクセスルールと ACF を使用した問題の解決方法を次に示します。

ここでは、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

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。SAP ASE 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。

次の例では、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。 SAP ASE 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。 初めにストアドプロシージャの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、SAP ASE 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

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、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、SAP ASE 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 SAP ASE サーバ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

ストアドプロシージャを作成して同時接続数を制限することができます。

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
```

ユーザ “bob” が SAP ASE 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> goCT-LIBRARY error:
ct_results(): network packet layer: internal net library error:
Net-Library operation terminated due to disconnect
```

このメッセージは、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

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できます。

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 時にサーバ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 時にサーバ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
```

月曜日に、ユーザ “bob” はセッションを作成します。

```
isql -Ubob -Ppassword
1> select 1
2> go
-----------
1
(1 row affected)
```

しかし、ユーザ “mark” 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
```

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

次のログイントリガ動作は、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、SAP ASE 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

- 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。 常にエラーログでエラーメッセージを確認してください。
- SAP ASE 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。
  SAP ASE 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 を有効にしてサーバ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、SAP ASE 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。
- サーバ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、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)。

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、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 execute 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

SAP ASE 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** または **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。

第 7 章 述部付きの権限

述部付きの権限にはローレベルの柔軟なアクセス制御の仕組みが含まれており、データアクセス時に SAP ASE によって評価される述部に基づいて、データに対する選択、更新、および削除の権限をさまざまなユーザ、グループ、役割に付与することができます。

いずれのデータローも述部で表された条件に一致しない場合は、そのローを結果セットで使用することはできません。

述部付きの権限が提供するデータプライバシーの保護は、データ内容やコンテキスト情報に応じて動的にユーザに権限を付与するローレベルのアクセス制御に基づいています。そのため、クライアントや Web サーバではなく、サーバにプライバシーポリシーを実装することができます。

述部は、テーブル、SQL 関数、組み込み関数など、他のオブジェクトにアクセスする場合があります。 こうした他のオブジェクトへのアクセスは、述部の所有者 (述部付きの権限の付与者) のパーミッションや役割と照合されます。 付与の対象となるオブジェクトに対して **select** コマンド、**update** コマンド、または **delete** コマンドを実行するユーザには、述部で参照されるオブジェクトに対する明示的なパーミッションは必要ありません。

述部付きの権限を使用すると、サービスプロバイダが 1 つのデータベースにデータを保存し、同じテーブルを複数の顧客で共有できるので、顧客ごとに個別のビューや **instead of** トリガを用意せずに済みます。 ユーザに直接付与された述部付きの権限、またはユーザのグループまたは役割によって間接的に付与された述部付きの権限に従って、個々のユーザについてデータがフィルタされます。

アクセスルールと述部付きの権限の違い

15.7, ESD #2 より前のバージョンの SAP ASE で提供されるアクセスルールと比較して、述部付きの権限の利点を説明します。

	アクセスルール	述部付きの権限
述部の範囲	ルールがバインドされたカラムのみを参照できる。 複数のカラムを参照するには、複数のアクセスルールが必要である。	テーブル内の任意のカラムを参照することも、sub-select を使用して、現在のデータベースまたは他のデータベース内の他のテーブルや関数を参照することもできる。

	アクセスルール	述部付きの権限
文の範囲	すべてのルールが **select**、**update**、**delete** に適用される。 アクセス権の種類ごとに異なる制限を適用できない。	どの述部をどのアクセス権に適用するかを付与者が指定する。 たとえば、ローの選択よりもローの更新や削除により厳密なアクセス制御を適用できる。
述部の結合	**and** または **or** を使用して個々のルールを結合できるが、優先度の分類を表すことはできない (例: **((*rule1* and *rule2*) or *rule3*)**)。	述部付きの権限では、任意の複雑さを持つブール式を構成できる。 適切に定義されたルールに基づいて、同じオブジェクトに対する複数の権限を結合できる。
範囲を対象に制限	**where name = *user*** のように、特定の対象に対する制限をルールの一部として表す必要がある。	管理者が複雑なロジックを述部に組み込まなくても、選択したユーザ、グループ、役割に述部を適用できる。
パーミッションシステムとの統合	SQL の権限とは分離されている。 ユーザがどのようにして特定の操作を禁止されているかを理解するには、ルールの分析が必要である。	パーミッションシステムによって強力に実施される。 アプリケーション開発者は、診断コマンドを使用して、述部によってクエリがどのように変更されるかを予測できる。

述部付きの権限に使用するコマンド

述部付きの権限の付与と取り消しには、次のコマンドを使用します。

- **grant ... where - where** 句で表される条件に基づいて、グループ、役割、個々のユーザにローレベルのアクセス制御を付与します。
- **grant role ... where - where** 句の評価に応じてアクティブ化される役割を付与します。
- **revoke ... [with {*pred_name* | {all |no} predicates}]** - revoke predicated privileges.
- **set show_transformed_sql** - ローレベルのフィルタリングのために追加された述部を含め、SQL テキスト全体を表示します。

『リファレンスマニュアル: コマンド』を参照してください。

述部付きの権限を使用するための SAP ASE の設定

述部付きの権限を使用するように SAP ASE を設定します。

1. SAP ASE Security and Directory Services に必要な ASE_SECDIR ライセンスを有効にします。
2. **enable predicated privileges** 設定パラメータを設定します

3. **permission cache entries** の値を設定します。 複数の述部付きの権限によって制限される 1 つのテーブルへのアクセス権を付与すると、必要な保護キャッシュエントリの数が増加します。 複数の述部の評価を必要とするアクセス権については、余分な保護キャッシュエントリを 1 つ許可します。
4. プロシージャキャッシュのサイズを大きくするかどうかを検討します。 ストアドプロシージャを実行するユーザがそれぞれ、プロシージャで参照されるテーブルに対して異なる述部付きアクセス権を持っている場合、プロシージャを実行するユーザごとに異なるプランが作成されます。 この場合、プロシージャキャッシュのサイズを大きくしなければならないことがあります。

enable predicated privileges

enable predicated privileges は、述部付きの権限を有効または無効にします。

概要情報	
デフォルト値	1 (オン)
有効な値	0 (オフ)、1 (オン)
ステータス	
表示レベル	包括
必要な役割	システム管理者
設定グループ	

enable predicated privileges は、述部付きの権限を有効または無効にします。

述部付きの権限の付与

grant を使用すると、**where** 句で指定した条件に基づいて、テーブルに対するローレベルのアクセス制御を付与できます。

構文は次のとおりです。

```
grant {all [privileges] | permission_list}
on table_name [correlation_name]
[(column_name_list)]
[where search_conditions
   [as pred_name]]
to {public | name_list | role_list}
[with grant option]
```

『リファレンスマニュアル: コマンド』を参照してください。

テーブルの **grant ... where** 文を使用すると、そのテーブルに対する **select** コマンド、**update** コマンド、または **delete** コマンドの条件を定義して、結果セットのローを

限定したり、除外したりすることができます。 *search_conditions* は、**select**、**update**、または **delete** で指定された **where** 句と連動して、ローフィルタとして機能します。 複雑な SQL の条件を含み、関数、テーブル、SQL 関数を参照する **search_conditions** を構成できます。 ただし、述部付きの権限でビューを参照することはできません。

次の例は、自分の給与と部下の給与を表示するためのパーミッションをエンジニアに付与します。

```
grant select on employee (salary)
  where ename = USER or
  emgr = USER to eng_role
```

この文は、**select** コマンド (**select into**、**insert select**、**update from** などの複合コマンドの一部である **select**、ビューでラップされた **select** コマンドを含む) のほか、**where** 句で `salary` カラムを参照する **select** コマンド、**update** コマンド、**delete** コマンドでもローをフィルタします。 述部は事実上、従業員が別の従業員の給与を表示できないよう制限します。

SAP ASE は、テーブルに対するクエリが処理されるときに **grant** 文の **where** 句を適用し、クエリが実行されるときに条件を評価します。

データを選択するためのアクセス権の付与

grant select コマンドの **where** 句は、**select** コマンド内のエンドユーザの **where** 句と結合して、**select** 操作から返されたローを限定します。

選択された個々のカラムに関係なくローをフィルタするには、カラムリストなしでテーブルの **grant ... where** 文を使用します。 次の **grant** 文は、自分が管理するプロジェクトの情報のみへの読み込みおよび書き込みアクセスを `project_leaders` に許可します。

```
grant select on projects
where user_id() = p_mgrid
to proj_leader_role
```

select が特定のカラムにアクセスする場合にローをフィルタするには、**grant** コマンドでカラムまたはカラムリストを使用します。 次の **grant** 文は、すべてのエンジニアの名前と部門へのアクセスをすべてのエンジニアに許可しますが、エンジニアの電話番号へのアクセスはマネージャのみに制限します。

```
grant select on employee(ename, edept)
  to eng_role
```

```
grant select on employee(ephone)
  where user_id() = emgr to eng_role
```

データを更新するためのアクセス権の付与

grant update コマンドの **where** 句は、**select** コマンド内のエンドユーザの **where** 句と結合して、**update** 操作の対象となるローを限定します。

ローをフィルタして特定のカラムのみを更新するには、**grant ... update** 文でカラムまたはカラムリストを使用します。 次の例は、管理部門および医療部門のすべてのスタッフに患者の住所の更新を許可しますが、医療ステータスの変更は患者の担当医のみに制限します。

```
grant update on patients (address, phone)
  to public
grant update on patients (medical_status)
  where USER = primary_md
  to doctor_role
```

doctor_role を持つユーザ以外が患者の医療ステータスを更新しようとすると、**update** コマンドでパーミッションエラーが発生します。 また、このコマンドでは、医師が更新できるローとカラムも制限されます。

データを削除するためのアクセス権の付与

grant delete コマンドの **where** 句は、エンドユーザの **where** 句と結合して、削除されるローを限定します。

次の例は、売上データを削除するためのパーミッションを Bob に付与します。

```
grant delete on employee
where edept = 'sales'
to Bob
```

Bob が次の文を実行します。

```
delete employee
where status = 'terminated'
```

この場合、SAP ASE の内部では次のような文が実行されます。

```
delete employee
where status = 'terminated' and edept = 'sales'
```

述部付きの権限を使用したデータプライバシーポリシーの実施

grant 文を使用して、従業員にプライベートなデータの表示を禁止するポリシーを実施することができます。

次の例の **grant** 文は、従業員に自分以外の給与の表示を禁止し、マネージャに自分のグループメンバの給与の表示を許可し、12 月以外は給与の更新を HR 部門に制限するポリシーを実施します。

```
grant select on employee as e (esalary)
  where e.ename = USER
```

```
  or USER in
   (select username(mgrid) from depts d
  where d.deid = e.eid)
  to public
grant update on employee (esalary)
  where datepart(month, current_date()) <> 12
  to hr_role
```

述部付きの権限の取り消し

テーブルに対するローレベルのアクセス権を取り消すには、revoke コマンドを使用します。

構文は次のとおりです。

```
revoke {all [privileges]
    | [all] permission_list}
on table_name (column_list)
[with { pred_name | {all |no} predicates}]
from {public | name_list | role_list}
```

- *pred_name* とともに *column_list* を使用すると、指定カラムについて述部付きのローレベルアクセス権が取り消されます。 このローレベル権限で参照されるカラムがほかにある場合、権限とそれに関連する指定述部は sysprotects に保持されます。 下記の **revoke** コマンドの例では、パーミッションが最初に次のように付与されているものとします。

  ```
  grant select on t1(col1,col2,col3)
    where col4 > 99 as pred1
    to user1
  ```

  次の例は、pred1 付きの t1.col2 に対する **select** パーミッションを取り消します。

  ```
  revoke select on t1 (col2)
    with pred1
    from user1
  ```

  ただし、user1 が t1.col3 を選択した場合は、pred1 が引き続き適用されます。
  さらに、付与者が次の文を発行したとします。

  ```
  revoke select on t1 (col1, col3)
    with pred1
    from user1
  ```

  または

  ```
  revoke select on t1 with pred1
  ```

この場合 pred1 を使用する t1 に対するすべてのパーミッションが user1 から取り消されます。

- すべての述部を取り消す場合、述部を指定する場合、または特定のアクセス権について述部なしの付与のみを指定付与対象者から削除する場合は、**with** 句を使用します。 **with** 句を省略した場合、述部付きの付与と述部なしの付与の両方が取り消されます。 次の例は、user1 に付与された述部付きのローレベル権限をすべて取り消します。

```
revoke select on t1 with all predicates
  from user1
```

次の例は、**with** 句を省略し、user1 に付与されたすべての述部なしの **select** アクセス権の述部なしの付与を取り消します。

```
revoke select on t1 with no predicates
```

『リファレンスマニュアル: コマンド』を参照してください。

SAP ASE における述部付きの権限の sysprotects への保存の仕組み

述部がない **grant** コマンドの場合、付与対象者に関する付与された権限の優先度の規則では、上位の **grant** が下位の **grant** を削除できると指定されています。 ただし、優先度が高い **grant** が述部付きの場合、通常、優先度が低いローが sysprotects に保持されます。

次の例では、public への 2 番目の述部なしの **grant** によって、個人 user1 への最初の **grant** が削除されます。

```
grant select on t1 to user1
grant select on t1 to public
```

ただし、述部付きの grant では、優先度が低い grant を保持できます。 たとえば、下記の user1 への最初の **grant** は全般的ですが、t1 に対する public への 2 番目の **grant** は条件付きです。 付与対象者 public の方が優先度が高くても、user1 への **grant** は削除されません。

```
grant select on t1
  to user1
grant select on t1
  where col1 = 4 to public
```

次の例では、2 番目の **grant** が追加されますが、権限は、col1 を除く t1 内のすべてのカラムに対する条件付きアクセス権として記録されます。

```
grant select on t1 (col1)
  to user2
grant select on t1
```

```
  where col1 = 4
  to user2
```

最初の **grant** は、カラム `col1` に対する条件なしのアクセス権を `user2` に付与します。これは、2 番目の **grant** の `col1` に対する条件付きアクセス権よりも強力であるため、削除されません。

述部付きの役割のアクティブ化

特定の条件に従って役割のアクティブ化を制限するには、**grant role ... where** コマンドを使用します。

アクティブな役割のみがアクセス権限を取得できます。 アクティブ化述部を持つ役割の付与対象者は、ユーザまたはログインプロファイルです (ログインプロファイルは、割り当てられたユーザにログイン時に適用される属性のテンプレートです)。

構文は次のとおりです。

```
grant role role_name
        [where pred_expression]
    to {username | login_profile_name }
```

where*pred_expression* 句は、役割のアクティブ化述部、つまり、指定された役割をアクティブ化する際に満たす必要がある SQL の条件です。 ユーザまたはログインプロファイルに役割を付与するには、*pred_expression* を使用します。

注意： アクティブ化述部を持つ役割を付与するには、`master` データベースを使用している必要があります。

役割のアクティブ化述部は、述部付きの権限と同様に、テーブル、ビュー、SQL 関数、組み込み関数などのデータベースオブジェクトにアクセスできます。 これらのアクセスは、述部の所有者 (役割の付与者) のパーミッションや役割と照合されます。set role 文を実行するユーザには、述部がアクセスするオブジェクトに対する明示的なパーミッションは必要ありません。

注意： アクティブ化述部ではビューを参照できますが、ローフィルタリング述部でビューを参照することはできません。

次の例は、オフピーク時におけるサーバのメンテナンス作業の実行をログイン “Bob” に許可します。 Bob が午前 8 時から午後 6 時の間に `oper_role` を有効にしようとすると、エラーが返されます。

```
grant role oper_role
  where datepart(hour, current_time())
  not between 8 and 18
  to Bob
```

ユーザまたはログインプロファイルに対する **grantrole** でアクティブ化述部を使用した場合、次の時点で述部が評価されます。

- 役割がユーザのデフォルトの役割として指定されている場合、ログイン時
- 役割がログインプロファイルに付与され、自動的にアクティブ化される役割として追加されている場合、ログイン時
- ユーザが役割を設定しようとしたとき

ログインプロファイルに付与された役割は、そのプロファイルに割り当てられているすべてのユーザに適用され、述部の評価に基づいて、ログイン時に自動的にアクティブ化されるように指定することができます。 たとえば、次の例では、`admin` ログインプロファイルに関連付けられているユーザが **resource_monitor** アプリケーションを使用するときには、sa_role が付与されるようにすることができます。

```
grant role sa_role
  where
    (select get_appcontext('SYS_SESSION', 'applname'))
    = 'mon_resource'  to loginprof_admin
```

役割のアクティブ化述部が false と評価された場合、**set role** コマンドに関するエラーが返されるか、ログイン時にデフォルトの役割や自動的にアクティブ化される役割が設定されません。

ローレベル権限を適用するための述部の結合

SAP ASE は、特定のテーブルに対する **update** コマンド、**delete** コマンド、**select** コマンドに述部を結合および追加します。

これは、次の点を考慮して行われます。

- 複数の付与者 - 1 人以上の付与者からの同じアクセス権について、関連するすべてのカラムにわたって特定の付与対象者、またはそのグループや役割に **grant** コマンドが適用されます。
- 複数のカラムセット - 1 人以上の付与者からの同じアクセス権について、さまざまなカラムセットにわたって特定の付与対象者、またはそのグループや役割に **grant** コマンドが適用されます。
- 異なるアクセス権 - 個々のアクセス権について同じオブジェクトに対する権限が付与されます。

特定のユーザアクセスについてユーザレベルまたはグループレベルで **revoke** ローが存在する場合、集約された付与は、ユーザの役割のセットに適用される述部で構成されます。

SAP ASE は、述部を結合する際に一連のルールに従います。

例
以下の例は、次のテーブルを作成することを前提としています。

```
create table employee (
   eid int not null,
   ename char(8) not null,
   eaddr varchar(20) not null,
   ecity char(10) not null,
   ezip char(5) not null,
   ephone int not null,
   esalary money not null,
   edept char(10) not null
)
```

さらに次の値を挿入します。

```
insert employee
values (1001, 'Bill','6 Tree St','Baytown','97364',3973465,100000,
'eng')
insert employee
values (1009, 'Sally', '2 Sun Dr', 'Henton', '97821', 1239786, 50000,
'sales')
insert employee
values (1005, 'Drew','7 Inn Rd','Denville','29052',4960034,110000,
'payroll')
insert employee
values (1010, 'Navid', '1 Bat Ave', 'Busitown', '60734',
3094509,175000, 'hr')
insert employee
values (1004, 'Pat', '5 Dot Ln', 'Toytown', '33109', 4247184, 90000,
'eng')
insert employee
values (1006, 'Sue', '4 Tip St', 'Uptown', '76420', 5258641, 200000,
'sales')
```

例 1: 同じテーブルアクセスについての複数の付与の述部の結合

ユーザコマンドで参照されるカラムにすべての述部が適用される場合、同じテーブルアクセスについての複数の付与の述部は、ブール演算子 **or** を使用して結合されます。

同じアクセス権について同じオブジェクトに対する複数の述部付きの権限が直接、または役割などによって間接的にユーザに付与されている場合、個々の **grant** コマンドの述部は、**or** パラメータを使用して結合されます。 1 つのアクセス権について述部付きの付与が追加されると、通常、ユーザに表示されるローが増えます。

たとえば、販売部門のマネージャであるユーザ Ben に、`employee` テーブルに対する次のようなパーミッションが付与されているとします。

```
grant select on employee (esalary)
  where edept = "sales"
  to sales_mgr_role
```

```
grant select on employee (eid, ename, eaddr,
  ezip)
  where ezip like "97%"
  to ben
```

アクティブな **sales_manager_role** を持つ Ben が次の文を実行した場合、カラム `eid` および `ename` は 2 つの述部付きの **grant** コマンドの影響を受けます。

```
select eid, ename from employee
```

SAP ASE は、内部で次の文を実行します。

```
select eid, ename from employee
  where edept = "sales" or
  ezip like "97%"
```

その後、SAP ASE は次の値を返します。

```
eid       ename
-----      ------
1001      Bill
1009      Sally
1006      Sue
```

例 2: さまざまなカラムセットに適用される複数の付与の述部の結合

さまざまなカラムセットに適用される複数の付与の述部は、ブール演算子 **or** と **and** を使用して結合されます。

結果セットに対する一連の述部による影響が均一でない場合、SAP ASE は、ブール演算子 **or** を使用して、各カラムに対する特定のアクセス権のすべての述部を結合します。 これらのブール式は、ブール演算子 **and** を使用して、すべてのカラムにわたって結合されます。

たとえば、例 1 と同じ権限が Ben に付与されているとします。ただし、最初の **grant** は従業員カラムのサブセットに影響します。

```
grant select on employee (esalary)
  where edept = "sales"
  to sales_mgr_role

grant select on employee (eid, ename, eaddr,
  ezip)
  where ezip like "97%"
  to ben
```

Ben が次の文を実行した場合、SAP ASE は、カラム `ename` および `esalary` が異なる **grant** コマンドの影響を受けることを認識します。

```
select ename, esalary from employee
```

SAP ASE は、内部で次の文を実行します。

```
select ename, esalary from employee
  where edept = "sales" and ezip like "97%"
```

その後、SAP ASE は次の値を返します。

```
enameesalary
sally50000
```

例 3: 異なるアクセス権についての複数の付与の述部の結合

異なるアクセス権についての複数の付与の述部は、ブール演算子 **and** を使用して結合されます。

1 つの述部を使用して 1 つのオブジェクトに対する **select access** が対象に付与され、別の述部を使用して update アクセスが許可されており、両方の述部付きの付与がアクセスに必要な場合、2 つの述部は、ブール演算子 **and** を使用して結合されます。

たとえば、最初の **grant** によって、郵便番号が「97...」の地域に住んでいる従業員の表示がカリフォルニアの管理者に許可されるとします。 2 番目の **grant** では、給与が年間 $100,000 未満の従業員のみの更新がこれらの管理者に許可されます。

```
grant select on employee
  where ezip like "97%"
  to calif_admin

grant update on employees (esalary)
  where esalary < $100000
  to calif_admin
```

`calif_admin` の役割を持つユーザが次の文を実行します。

```
select eid, ename, esalary from employee
```

SAP ASE は次の値を返します。

```
eid     ename     esalary
----     ------    -----------
1001       Bill     100,000.00
1009      Sally      50,000.00
```

`calif_admin` の役割を持つユーザが次の文を実行します。

```
update employees
set esalary = esalary + (esalary * 0.05)
```

SAP ASE は、内部で次の文を実行します。

```
update employees
set esalary = esalary + (esalary * 0.05)
  where ezip like "97%"
  and esalary < $100000
```

その後、SAP ASE はローを次のように更新します。

```
eid     ename     esalary
1009    Sally      $52500
```

述部付きの権限がある場合の SQL の動作

次の **set** パラメータとシステムプロシージャを使用して、述部付きの権限に関する情報を表示します。

次の set パラメータは、述部付きの権限に関する情報を表示します。

- **set show_transformed_sql** コマンドは、ローレベルのフィルタリングのために追加された述部を含め、SQL クエリ全体を表示します。 **set noexec on** とともに使用すると、ユーザ、グループ、およびアクティブな役割に関連する述部が追加された後、クエリを実行することなく、最終的な SQL クエリを検証することができます。
- **set show_permission_source** は、付与対象者、付与対象者のタイプ、付与者、アクション、オブジェクト、および述部を表形式で表示します。

『リファレンスマニュアル: コマンド』を参照してください。

次のシステムプロシージャは、述部付きの権限に関する情報を表示します。

- **sp_helptext** は述部のテキストを表示します。 **sp_helptext** では、述部のユーザ定義名 (ある場合) または内部名を指定してください。

  ```
  sp_helptext pred1
  # Lines of Text
  ---------------
  1
  text
  ------------------------------------------------------------
  grant select on tab1 where col1 = 5 as pred1 to robert
  ```

  sp_helptext は、**sp_hidetext** によって隠されていない結果のみを表示します。次の例は、**sp_hidetext** によって隠されたソーステキストとともに述部を表示します。

  ```
  Msg 18406, Level 16, State 1:
  Procedure 'sp_helptext', Line 313:
  Source text for compiled object pred1 (id = 592002109) is hidden.
  ```

- **sp_helprotect** は、述部付きの権限がある場合、その名前を述部カラムに表示します。

『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。

述部付きの権限に対する所有権の連鎖の影響

呼び出し元のオブジェクトとテーブルを同じユーザが所有している場合、ストアドプロシージャ、SQL 関数、ビュー、またはトリガと、そのオブジェクトがアクセスするテーブルとの間に所有権の連鎖が存在します。

この所有権の連鎖によって、ストアドプロシージャ、ビュー、SQL 関数、またはトリガの所有者がプロシージャやビューなどに対する明示的なアクセス権を別のユーザに付与することで、従属オブジェクトに対する暗黙的なアクセス権がそのユーザに与えられます。 従属オブジェクトに対するアクセス権を付与する必要はありません。

所有権の連鎖によってユーザがテーブルに対する暗黙的なパーミッションを持っている場合、そのアクセス権にはローフィルタリングのための **grant** コマンドの述部は適用されません。

たとえば、employee テーブルの所有者が、給与部門の従業員データを表示するためのアクセス権をユーザ “Priscilla” に付与するとします。

```
grant select on employee
  where dept = 'payroll'
  to priscilla
```

Priscilla が次のように入力します。

```
select name, phone from employee
  where city='SFO'
```

サンフランシスコに住んでいて給与部門で働いている従業員の名前と電話番号だけが返されます。

ただし、`employee` テーブルの所有者が次のプロシージャを作成し、`execute` アクセス権を Priscilla に付与した場合は、`employee` と `employee_addresses` 間の所有権の連鎖が検出されます。

```
create procedure employee_addresses as
select name, phone from employee
```

このプロシージャのコンテキストでは、`employee` テーブル内のすべてのローへの暗黙的なアクセス権が Priscilla に与えられます。前の **grant** の述部による制限は適用されません。

ansi_permissions と述部付きの権限

update コマンドまたは **delete** コマンドに、更新または削除されるローを持つテーブル内のカラムを参照する **where** 句が含まれる場合、SAP ASE は、そのセッショ

ンで **ansi_permissions** が有効なときにのみ、これらのカラムに対する `select` パーミッションがあるかどうかを確認します。

SAP ASE は、次のいずれの場合に該当するかに応じて **update** 文と **delete** 文の述部付きの権限を適用します。

- **ansi_permissions** が有効な場合 - テーブルの所有者が **where** 句でカラムに対する述部付きの **select** アクセス権を付与している場合、grant **update** 文で指定された述部に加えて grant select の述部付きの権限が適用されます。 両方の述部を適用することで、ローセットが予期せず制限されないことを確認してください。
- **ansi_permissions** が無効な場合 - **where** 句で参照される、更新または削除されるテーブルのカラムに対する **select** パーミッションは必要ありません。 テーブルに対する **select** アクセス権に付与された述部は、**update** 文や **delete** 文には適用されません。

次の例では、**ansi_permissions** が述部付きの権限にどのように影響するかについて説明します。

- 最初の **grant** では販売部門の従業員の給与の更新をユーザ “Bob” に許可し、2 番目の **grant** では直属の部下である従業員のみの従業員 ID と住所の選択を Bob に許可します。

  ```
  grant update on employee (salary)
    where edept = "sales"
    to bob

  grant select on employee (eid, eaddress)
    where user_name(mgrid) = USER
    to bob
  ```

  Bob が `eid` に基づいて従業員の給与を更新しようとすると、上記の例の **grant** コマンドの一方または両方の述部が適用されます。

  ```
  update employee
    set salary = salary * 0.1
    where eid = 1006
  ```

- セッションで **set ansi_permissions** が有効になっている場合、SAP ASE は **update** 文と **select** 文の両方に述部を追加してから **update** コマンドを実行します。

  ```
  update employee
    set salary = salary * 0.1
    where eid = 1006
    and edept = "sales"
    and user_name(mgrid) = USER
  ```

  Bob の述部付きの **select** パーミッションでは、直属の部下である従業員の ID のみの検索が許可されているため、Bob の述部付きの **select** アクセス権によって、更新または削除できるローが制限されます。
  ansi_permissions が無効な場合、SAP ASE は、次のような **update** コマンドを実行します。

```
update employee
  set salary = salary * 0.1
  where eid = 1006
  and edept = 'sales'
```

ansi_permissions が有効な場合と無効な場合の動作を要約すると、次のようになります。

- 無効 - Bob は、選択を許可されていないローを更新および削除できます。
- 有効 - **select** アクセス権の述部付きの付与における **where** 句によって、更新または削除できるローセットが限定されます。

注意： 上記の例の **grant** コマンドの述部の一方または両方が適用されます。

述部によるアクセスのパーミッション

述部付きの権限の付与者は、述部で参照されるテーブル、組み込み関数、または SQL 関数にアクセスするための権限を持っている必要があります。

ローフィルタリング述部については、SAP ASE は、ユーザの **select** コマンド、**update** コマンド、または **delete** コマンドによるアクセスを検証するときに、述部によるアクセスも実行時に同時に検証します。 役割のアクティブ化述部によるアクセスは、ユーザが役割をアクティブ化するときに検査します。 述部が追加されたコマンドを入力するユーザは、述部の参照オブジェクトに対する権限を持っている必要はありません。

たとえば、データプライバシーを確保するには、データ所有者のプライバシーポリシーを含むテーブルやカラムにアクセスする述部を指定する必要があります。これらのプライバシーポリシーへのアクセスは、少数のユーザに制限します。データのユーザは、ポリシーを実施するために使用される内部データを参照する権限を持っている必要はありません。

SAP ASE は、述部付きの権限によるアクセスのパーミッションを権限の付与者に対して強制する際に、グループやデフォルトの役割のメンバシップによる付与者の直接的および間接的なパーミッションを考慮します。 ユーザのデフォルトの役割には、自動アクティブ化を目的としてログインプロファイルによって付与された役割も含まれます。

プライバシーメタデータテーブルに対するパーミッションを持つアプリケーションセキュリティの役割を作成できます。 その後、付与の述部が必要なデータにアクセスできるように、デフォルトの役割または自動的にアクティブ化される役割としてその役割をオブジェクト所有者に割り当てることができます。

次の例では、`purchases` テーブルの所有者である Bob が、`market_role` を持つ市場アナリストに、データを共有することにした顧客の商品購入情報を表示する

ためのパーミッションを付与します (顧客の設定は privacy_db に保存されます)。

```
grant select on purchases p
  where exists (select 1 from privacy_db..choices c   where p.custid
= c.id)
  to market_role
```

market_role を有効化したユーザ Alice が purchases テーブルから選択する際、次のようなアクセス検査が実行されます。

- purchases テーブルに対する Alice の直接的または間接的な **select** パーミッション
- privacy_db..choices テーブルから選択するための Bob の直接的または間接的なパーミッション

注意： 述部付きの **grant** が適用されるオブジェクトの所有権が **alter .. modify owner** コマンドによって変更された場合、述部付きの権限の所有権が変更されます。 述部がアクセスするオブジェクトに対するパーミッションが新しい所有者に付与されていない場合、述部の適用によって実行時エラーが発生します。

述部付きの権限とトリガ

SAP ASE は、疑似テーブルにはローフィルタリング述部を適用しません。 ベーステーブルに適用されたローフィルタリング述部によって、inserted テーブルおよび deleted テーブル内のデータが制限されます。

通常のテーブルに対するトリガによるアクセスには、述部によるローフィルタリングが適用されます。

述部付きの権限の再コンパイル

述部付きの権限によって制御されるテーブルにストアドプロシージャ、トリガ、SQL 関数がアクセスする場合、SAP ASE は、異なるユーザによるアクセスだけでなく、1 つ以上の役割をアクティブ化している同じユーザによるアクセスについても (プランが最後にコンパイルされたときにアクティブ化されていた役割とはアクティブな役割が異なる場合)、プロシージャオブジェクトを再コンパイルしなければならないことがあります。

同じプロシージャを実行する個々のユーザについて常に再コンパイルしなくても済むように、SAP ASE は、現在のユーザに適した「プロファイル」があるキャッシュされたプランを選択しようとします。 たとえば、プロシージャによるアクセスが role1、role2、role3 に対する異なる述部付きの付与によって制御されているとすると、role1 と role2 がアクティブな Joe による実行の際にプロシー

ジャが最初にコンパイルされた場合、role2 と role3 がアクティブな Bob について、同じプロシージャが再コンパイルされる可能性があります。

Bob と Joe についてコンパイルされたプランがプロシージャキャッシュに含まれている場合、コンパイルしなければならない可能性が低いプランのコピーが選択されます。

述部を使用するプランをキャッシュから選択する際にはまず、ユーザとプランの保護「プロファイル」で次の情報を照合します。

- プランがコンパイルされたユーザ ID。 ユーザ ID がプランの保護プロファイルに保存されるのは、プロシージャがアクセスするテーブルに対する 1 つ以上の述部付きの権限がユーザに直接付与されている場合のみです。
- グループ ID (1 つ以上の述部付きの権限がグループに付与されている場合)。
- 役割 ID のリスト (1 つ以上の述部付きの権限が役割に付与されている場合)。

ユーザのプロファイルと一致するプランを選択しても、そのプランが最新であり、現在のユーザのアクセスに必要な述部を含んでいるとはかぎりません。 SAP ASE は、プロシージャ内の **select** コマンド、**update** コマンド、**delete** コマンドがそれぞれ、現在のユーザの述部付きの付与を表す **where** 句を使用してコンパイルされていることを検証します。

そうでない場合は、プロシージャを再コンパイルする必要があります。 述部付きの権限には、実行段階におけるプランの再コンパイルの新しい意思決定ポイントとして保護検査が導入されています。

プロシージャキャッシュを効率的に使用し、ストアドプロシージャプランを共有するには、役割ベースのプライバシーポリシーを使用する必要があります。 ストアドプロシージャで参照されるオブジェクトへの述部付きのアクセス権は、エンドユーザがアクティブ化できる少数の役割に付与してください。

述部の再帰的処理の禁止

ローフィルタリング述部または役割のアクティブ化述部で行われるアクセスが述部付きの付与によって制御されている場合、SAP ASE はエラーを返します。 たとえば、次の場合、アクセス検査の再帰的サイクルが発生します。これは禁止されています。

- table1 が、table2 を参照する **predicate1** 付きの **grant** によって保護されている。
- table2 が、table1 を参照する **predicate2** 付きの **grant** によって保護されている。

役割のアクティブ化述部にも同様の制限があります。 たとえば、次の場合、ユーザーが **set role** r1 を実行しようとすると、エラーが返されます。

- `table1` が述部付きの **grant** によって保護されている。
- `r1` に対する **grant** がアクティブ化述部で `table1` を参照している。

ログイン時、set proxy の実行時、付与者に代わって述部付きのアクセス権を評価するときなど、ユーザのデフォルトの役割を適用するときに、SAP ASE は、アクティブ化述部で付与されているデフォルトの役割を無視します。

述部による情報の漏洩

述部付きの権限の適用によって、データを表示する権限がないユーザにテーブル内のローに関する情報が漏洩したり、データの一貫性が損なわれたりする可能性があります。

たとえば、従業員 ID は機密でユニークであり、販売部門の従業員の ID のみを処理する権限がユーザ Joe に与えられているとします。この場合、Joe が販売部門の従業員の ID を更新する際、SAP ASE が一意性違反メッセージを発行するかどうかによって、特定の ID がテーブル全体に存在するかどうかについての確認を行うことができます。

更新可能なビューにも、同様の情報漏洩の可能性があります。 述部付きの権限によるアクセス制御を利用するアプリケーション設計者は、プライマリキーの更新を禁止するとともに、参照制約があるテーブルに対する述部付きの **update** および **delete** 権限を持つユーザを制限する必要があります。

次の例では、カラム `cust_name` は、`customer.name` の値によって制約されるテーブル `t_orders` の外部キーです。

```
create table t_orders (ordernum int, orders_dt date,  cust_name
char(60) references customer.name, state char(2))
```

受注部門のある従業員は、アイオワ州の顧客に出荷する受注について `t_orders` テーブルに対する **select** および **update** パーミッションを持っています。

```
grant select, update on t_orders
where state = 'IO'
to procure_role
```

受注部門のこの従業員は、この企業の国全体の顧客をリストする顧客テーブルへのアクセス権は持っていません。 ニューヨークの特定の顧客がこの企業の商品を購入しているかどうかを調べるために、 ニューヨークの顧客の顧客名を使用し、次のコマンドを入力して、アイオワ州で行われた受注を更新します。

```
update orders set cust_name = 'Ronald Crump'
where ordernum = 345
```

上記の文が外部キー制約違反を返さなかった場合、Ronald Crump がこの企業の顧客であることがわかります。

第 8 章　細密なパーミッション

細密なパーミッションを使用してシステム権限を付与すると、サイト固有の役割を要件に一致する権限で構成したり、システム管理者やデータベース所有者によるユーザデータへのアクセスを制限したりすることができます。

付与可能なシステム権限により、「作業の分割」(特定の操作セットにおいて、セット内のすべての操作の実行をいずれの個人にも許可しないようにする必要がある) や「最小権限」(情報システムのすべてのユーザにジョブの実行に必要なだけの権限を付与する必要がある) を適用できます。

ある権限から別の権限を取り消したり、ある権限を別の権限に付与したりすることはできません。 ただし、付与対象者が実行できる内容が権限間で重複しても構いません。ある権限を所有すると、さらに細密な別の権限を所有することになる場合があります。

細密なパーミッションを有効にすると、システム標準の役割 (sa_role、sso_role、oper_role、および replication_role) が、明示的に付与された権限のセットで構成される権限コンテナとして再構成されます。 明示的に付与されたシステム権限をシステム標準の役割から取り消し、それらの権限をシステム標準の役割に再付与することができます。

細密なパーミッションを使用するための SAP ASE の設定

細密なパーミッションを使用するように SAP ASE を設定します。

1. Security and Directory Services ライセンス、ASE_SECDIRS を有効にします。
2. **enable granular permissions** 設定パラメータを有効にし (sso_role が必要)、ASE_SECDIRS ライセンスをチェックアウトします。 **enable granular permissions** を無効にするには、細密なシステム権限 **manage security configuration** が必要です。この設定パラメータを無効にすると、ライセンスが再びチェックインされます。

enable granular permissions を有効にする場合、SAP ASE は、ロックされていない少なくとも 1 つのアカウントに `manage server permissions` 権限があること、ロックされていない少なくとも 1 つのアカウントに `manage security permissions` 権限があることを確認します。 **enable granular permissions** を無効にする場合は、sso_role を持つロックされていないアカウントがサーバに少なくとも 1 つあること、ロックされていない少なくとも 1 つのアカウントが sa_role を持っていることを確認します。

システム権限

細密なパーミッションによって、サーバワイドの権限とデータベースワイドの権限が定義されます。

サーバワイドの権限の付与または取り消しは、マスタデータベースで行う必要があります。 付与対象者は、マスタデータベース内の役割、ユーザ、またはグループでなければなりません。 サーバワイドの権限のパーミッション情報は `master.dbo.sysprotects` に保存されます。

データベースワイドの権限の付与または取り消しは、その権限を必要とするコマンドの対象となるデータベースで行う必要があります。 付与対象者は、そのデータベース内のユーザ、グループ、または役割です。 データベースワイドの権限のパーミッション情報は *database_name*`.dbo.sysprotects` に保存されます。

サーバワイドの権限とデータベースワイドの権限の付与または取り消しを行うには、**grant** コマンドまたは **revoke** コマンドを使用します。 たとえば、ユーザ Joe に任意のデータベースのダンプを許可するには、適切な権限を持つユーザがマスタデータベースから次のコマンドを発行します。

```
grant dump any database to joe
```

Joe 自身およびデータベース `db2` 内の他のユーザに代わって任意のオブジェクトの作成を Joe に許可するには、適切な権限を持つユーザが `db2` から次のコマンドを発行します。

```
grant create any object to joe
```

システム標準の役割に含まれる権限の影響

システム標準の役割に含まれる権限の付与および取り消しの影響と **enable granular permissions** パラメータについて説明します。

システム標準の役割に含まれる権限の付与や取り消しを行う際には、次の点に注意してください。

- システム標準の役割に明示的に付与されている権限をその役割から取り消すことができます。 権限を取り消すと、役割の所有者は、その権限に関連する操作をサーバで実行できなくなります。
- 明示的に付与されている権限を有効にするには、**enable granular permissions** 設定パラメータを有効にする必要があります。 **enable granular permissions** が無効な場合、システム標準の役割が持つ権限は、15.7 ESD #2 より前のバージョンの

SAP ASE と同じになります。 これらの権限は役割に暗黙的に付与され、取り消すことはできません。

変更したシステム標準の役割をデフォルトの権限の設定に戻すには、**sp_restore_system_role** を使用します。

パーミッションの管理

システム権限はすべて、`manage server permissions` 権限、`manage security permissions` 権限、または `manage database permissions` 権限を持つユーザが管理します。 オブジェクトパーミッションは、オブジェクト所有者または `manage any object permission` 権限を持つユーザが管理します。

manage security permissions 権限

`manage security permissions` 権限を持つユーザは、セキュリティ関連のサーバワイドの権限およびセキュリティ関連のデータベースワイドの権限の付与や取り消しを行うことができます。

表 14 : manage security permissions 権限で管理されるサーバワイドの権限

サーバワイドの権限
`change password`
`checkpoint (sybsecurity)`
`dump database (sybsecurity)`
`load database (sybsecurity)`
`manage any login`
`manage any login profile`
`manage any remote login`
`manage auditing`
manage roles
`manage security configuration`
`manage security permissions`
`online database (sybsecurity)`
`own database (sybsecurity)`

サーバワイドの権限
set proxy
use database (sybsecurity)

表 15 : manage security permissions 権限で管理されるデータベースワイドの権限

データベースワイドの権限
create encryption key
decrypt any table
manage any encryption key
manage column encryption key
manage database permissions
manage master key
manage service key
update any security catalog

manage security permissions は、新しくインストールされたサーバ上で初期設定で sso_role に明示的に付与されるため、sa アカウントはデフォルトで manage security permissions 権限を持っています。 sso_role から manage server permissions を取り消すと、この役割を持つユーザはセキュリティ関連の権限を付与したり、取り消したりすることができなくなります。

ユーザが予期せずサーバをロックしてしまうことがないように、SAP ASE は、manage security permissions 権限を持つロックされていないユーザアカウントがサーバに少なくとも 1 つあることを確認します。

manage server permissions 権限

manage server permissions 権限を持つユーザは、サーバワイドの権限の付与や取り消しを行うことができます。

表 16 : manage server permissions 権限で管理されるシステム権限

allow exceptional login
checkpoint any database
checkpoint (sybsecurity を除く任意のデータベース)

connect
create database
dbcc checkallock any database
dbcc checkcatalog any database
dbcc checkdb any database
dbcc checkindex any database
dbcc checkstorage any database
dbcc checktable any database
dbcc checkverify any database
dbcc fix_text any database
dbcc indexalloc any database
dbcc reindex any database
dbcc tablealloc any database
dbcc textalloc any database
dbcc tune
dump any database
dump database (sybsecurity を除く任意のデータベース)
kill
kill any process
load any database
load database (sybsecurity を除く任意のデータベース)
manage any database
manage any ESP
manage any thread pool
manage cluster
manage data cache
manage disk
manage dump configurations

manage lock promotion threshold
manage resource limit
manage server
manage server configuration
manage server permissions
mange execution classes
map external file
monitor server replication
mount any database
online any database
online database (sybsecurity を除く任意のデータベース)
own any database
own database (sybsecurity を除く任意のデータベース)
quiesce any database
set switch
347 = set tracing
set tracing any process
show switch
shutdown
unmount any database
use any database
use database (sybsecurity を除く任意のデータベース)

manage server permissions は、新しくインストールされたサーバ上で初期設定で sa_role に明示的に付与されます。 sa_role から manage server permissions を取り消すと、sa_role を持つユーザはサーバワイドの権限を付与したり、取り消したりすることができなくなります。

ユーザが予期せずサーバをロックしてしまうことがないように、SAP ASE は、manage server permissions 権限を持つロックされていないユーザアカウントがサーバに少なくとも 1 つあることを確認します。

manage database permissions 権限

`manage database permissions` 権限を持つユーザは、暗号化キー関連の権限 (manage security permissions 権限を持つユーザが管理) を除くデータベースワイドの権限の付与や取り消しを行うことができます。

`manage database permissions` 権限を持つユーザは、他のユーザおよび自分自身に対して `manage database permissions` 権限の付与や取り消しを行うことはできません。

データベースを作成しても、データベース所有者を含め、役割やユーザに `manage database permissions` 権限が初期設定で付与されるわけではありません。 データベースユーザがデータベースパーミッションの付与や取り消しを行えるようにするには、`manage security permissions` 権限を持つユーザが `manage database permissions` 権限をそのユーザに明示的に付与する必要があります。

表 17 : manage database permissions 権限で管理される権限

`alter any object owner`
`alter any table`
`create any default`
`create any function`
`create any index`
`create any object`
`create any procedure`
`create any rule`
`create any table`
`create any trigger`
`create any view`
`create default`
`create function`
`create procedure`
`create rule`
`create table`

create trigger
create view
dbcc checkallock
dbcc checkcatalog
dbcc checkdb
dbcc checkindex
dbcc checkstorage
dbcc checktable
dbcc checkverify
dbcc fix_text
dbcc indexalloc
dbcc reindex
dbcc tablealloc
dbcc textalloc
delete any table
drop any default
drop any function
drop any object
drop any procedure
drop any rule
drop any table
drop any trigger
drop any view
execute any function
execute any procedure
identity_insert any table
identity_update any table
insert any table

manage abstract plans
manage any object permission
manage any statistics
manage any user
manage checkstorage
manage database
manage replication
references any table
reorg any table
report checkstorage
select any audit table (sybsecurity のみで使用可能)
select any system catalog
select any table
setuser
transfer any table
truncate any table
truncate any audit table (sybsecurity のみで使用可能)
update any table

manage any object permission 権限

manage any object permission 権限を持つユーザは、任意のデータベースユーザが所有する任意のオブジェクトに対するオブジェクト固有のパーミッションの付与や取り消しを行うことができます。 オブジェクト所有者は、自分自身が所有するオブジェクトに対するすべての権限を付与できます。

表 18 : manage any object permission 権限で管理されるオブジェクトパーミッション

delete
delete statistics
execute

identity_insert
identity_update
insert
references
select
transfer table
truncate table
update
update statistics

システム標準の役割に付与される権限

システム標準の役割には、デフォルトで権限が付与されます。

表 19 : sa_role にデフォルトで付与される権限

allow exceptional login
checkpoint any database
connect
create database
dbcc checkalloc any database
dbcc checkcatalog any database
dbcc checkdb any database
dbcc checkindex any database
dbcc checkstorage any database
dbcc checktable any database
dbcc checkverify any database
dbcc fix_text any database
dbcc indexalloc any database
dbcc reindex any database

dbcc tablealloc any database
dbcc textalloc any database
dbcc tune
dump any database
kill any process
load any database
manage any database
manage any ESP
manage any execution class
manage any thread pool
manage cluster
manage data cache
manage disk
manage dump configuration
manage lock promotion threshold
manage resource limit
manage server
manage server configuration
manage server permissions
map external file
mount any database
online any database
own any database
quiesce any database
select on get_appcontext
select on list_appcontext
select on rm_appcontext
select on set_appcontext

set switch
set tracing any process
show switch
shutdown
unmount any database

表 20 : sso_role にデフォルトで付与される権限

alter any object owner (任意のデータベース)
change password
decrypt any table (任意のデータベース)
manage any encryption key (任意のデータベース)
manage any login
manage any login profile
manage any remote login
manage any user (任意のデータベース)
manage auditing
manage roles
manage security configuration
manage security permissions
select on authmech
show switch
set tracing any process
update any security catalog (任意のデータベース)

表 21 : oper_role にデフォルトで付与される権限

checkpoint any database
dump any database
load any database
manage dump configuration

online any database
use any database

表 22 : replication_role にデフォルトで付与される権限

checkpoint any database
dump any database
load any database
manage replication (任意のデータベース)
monitor server replication
online any database
quiesce any database
truncate any table (任意のデータベース)
truncate any audit table (sybsecurity)

表 23 : keycustodian_role にデフォルトで付与される権限

manage any encryption key

役割またはデータベース所有者をデフォルトの権限の設定に戻すには、**sp_restore_system_role** (manage security permissions 権限が必要) を使用します。

データベース所有者に割り当てられる権限

データベース所有者には、デフォルトで権限が付与されます。

表 24 : データベース所有者にデフォルトで付与される権限

alter any object owner
create default
create function
create procedure
create rule
create table

create trigger
create view
dbcc checkalloc
dbcc checkcatalog
dbcc checkdb
dbcc checkindex
dbcc checkstorage
dbcc checktable
dbcc checkverify
dbcc fix_text
dbcc indexalloc
dbcc reindex
dbcc tablealloc
dbcc textalloc
manage abstract plans
manage any user
manage checkstorage
manage database
manage database permissions (sybsecurity のみ)
manage replication
report checkstorage
select any audit table (sybsecurity のみ)
setuser
truncate any audit table (sybsecurity のみ)

データベース所有者には次のルールが適用されます。

- 新しく追加されたデータベースワイドの権限 (上記の権限は除く) がデフォルトでデータベース所有者に付与されることはありません。 追加の権限をデータ

ベース所有者に付与するには、明示的な **grant** コマンドを使用する必要があります。

- データベース所有者には、setuser 権限がデフォルトで明示的に付与されます。 データベース所有者が他のユーザになり代わることがないようにするには、データベース所有者から setuser 権限を取り消します。
- データベースに対する own any database 権限または own database 権限を持つユーザは、データベースの有効なユーザであるかどうかに関係なく、データベース所有者としてデータベースにログインします。 このユーザが作成したオブジェクトは、sysobjects.uid に **UID=1**、sysobjects.loginame にそのログイン名を保持します。 own any database と own database の両方の権限がこのユーザから取り消された場合、このユーザは、自分のユーザ ID を使用するか、データベースのユーザとして追加されていないときには guest として、データベースにログインします。

細密なパーミッションによる sa_serverprivs_role の追加

sa_serverprivs_role は、sa にデフォルトで付与されるユーザ定義の役割です。 **enable granular permissions** が有効な場合、システム管理者はこの役割によって、SAP ASE を実行するために必要なすべての権限を取得します。

ユーザ定義の役割である sa_serverprivs_role は、ログインに付与されていても、デフォルトでは、そのログインについてログイン時に自動的にアクティブ化されません。

ログイン時におけるこの役割の自動アクティブ化を有効にするには、**alter login** を使用します。

注意： sa_serverprivs_role を通常のユーザ定義の役割として使用しないことをおすすめします。

enable granular permissions を有効にすると、sa ログインは、データベースにおいてデータベース所有者と同じ権限を取得します。 sa ログインがデータベースワイドの他の権限を取得するには、自分自身をデータベースのユーザとして追加し、自分自身にそれらの権限を付与します。

表 25 : sa_serverprivs_role にデフォルトで付与される権限

allow exceptional login
change password
checkpoint any database

connect
create database
dbcc checkalloc any database
dbcc checkcatalog any database
dbcc checkdb any database
dbcc checkindex any database
dbcc checkstorage any database
dbcc checktable any database
dbcc checkverify any database
dbcc fix_text any database
dbcc indexalloc any database
dbcc reindex any database
dbcc tablealloc any database
dbcc textalloc any database
dbcc tune
dump any database
kill any process
load any database
manage any database
manage any ESP
manage any execution class
manage any login
manage any login profile
manage any remote login
manage auditing
manage cluster
manage data cache
manage disk

manage dump configuration
manage lock promotion threshold
manage resource limit
manage roles
manage security configuration
manage security permissions
manage server
manage server configuration
manage server permissions
map external file
mount any database
online any database
own any database
quiesce any database
select on authmech
select on get_appcontext
select on list_appcontext
select on rm_appcontext
select on set_appcontext
set switch
set tracing any process
show switch
shutdown
unmount any database

システム管理者に付与されるデフォルトの役割

ログイン sa には、デフォルトで次の役割が付与されます。

sa_role

sso_role
oper_role
mon_role
sybase_ts_role
sa_serverprivs_role

システム管理者およびデータベース所有者の権限の制限

システム標準の役割に付与された細密なパーミッションを役割から取り消して、その役割の権限を制限することができます。 一般的なユースケースについては、下記の例を参照してください。

例 1:

`manage server permissions` 権限を持つユーザは、`own any database` 権限と `manage server permissions` 権限を取り消すことによって、sa_role を持つユーザがユーザ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。

```
use master
revoke own any database from sa_role
revoke manage server permissions from sa_role
```

例 2:

データベース所有者には、デフォルトで `setuser` 権限が付与されます。そのため、データベース所有者は、他のユーザになり代わってそのユーザのデータにアクセスできます。

他のユーザのデータにアクセスできないようにするには、データベース所有者から `setuser` 権限を取り消します。 データベース所有者が自分自身に setuser 権限を付与できないようにするには、データベースで `manage database permissions` 権限がデータベース所有者に付与されていないことを確認します。 デフォルトでは、`manage database permissions` 権限はデータベース所有者に付与されません。

たとえば、データベース `db1` でデータベース所有者から `setuser` 権限を取り消すには、次のコマンドを発行します。

```
use db1
```

```
revoke setuser from dbo
```

`model` データベースでこれらの権限を変更して、これを、今後作成されるユーザデータベースのデフォルトの動作にします。

```
use model
```

```
revoke setuser from dbo
```

例 3：

sa_role ユーザは、サーバを誤って停止してしまうことがあります。 これを防ぐために、manage server permissions 権限を持つシステム管理者は、sa_role から shutdown 権限を取り消し、サーバオペレーションの停止を担当する管理者のみにその権限を付与することができます。

たとえば、ユーザ joe と bob (2 人とも sa_role を所有) に shutdown 権限を付与し、他のすべてのユーザからこの権限を取り消すには、manage server permissions 権限を持つユーザが次のコマンドを発行します。

```
use master
grant shutdown to joe, bob
revoke shutdown from sa_role
```

Enable granular permissions と sybsecurity

enable granular permissions を有効にした場合、次の制限が sybsecurity データベースに適用されます。

- manage auditing 権限を持つユーザのみが sybsecurity を作成できます。
- manage auditing 権限を持つすべてのユーザが、データベース所有者として sybsecurity にアクセスできます。
- sybsecurity のデータベース所有者には、デフォルトで manage database permissions 権限、select any audit table 権限、truncate any audit table 権限が付与されます。
- “any” という単語を含むすべてのサーバ権限が sybsecurity に適用されるわけではありません。 たとえば、own any database 権限を持っていても、データベース所有者として sybsecurity にアクセスすることはできません。
- manage security permissions 権限を持つユーザのみが、sybsecurity に対する own database、dump database、load database、checkpoint、online database の各権限を他のユーザに付与できます。

ロックアウトされたサーバへのログイン

enable granular permissions が有効な場合、change password 権限を持つすべてのログインがパスワードを忘れてしまったときにのみ、SAP ASE サーバがロックアウトされることがあります。 dataserver コマンドラインには、サーバのロックを解除するための次のパラメータがあります。

- -p *login_name* - サーバを起動する際にログイン名を指定して、そのアカウントのパスワードをリセットします。 このアカウントの新規パスワードとしてランダムパスワードが生成、表示、暗号化されて、master..syslogins に保存されます。 細密なパーミッションが無効な場合、*login_name* は sso_role を持っている必要があります。 細密なパーミッションが有効な場合、*login_name* は change password 権限を持っている必要があります。
- -u *login_name* - ロックを解除するログイン名を指定します。 細密なパーミッションが無効な場合、*login_name* は sso_role または sa_role を持っている必要があります。 細密なパーミッションが有効な場合、*login_name* は change password 権限を持っている必要があります。
- -A **system_role**、--role-logins - システム標準の役割の名前を指定して、その役割を持つログインアカウントのリストをログファイルに出力します。
- -n system_privilege、--permission-logins - システム権限の名前を指定して、そのシステム権限を持つログインアカウントのリストをログファイルに出力します。

ロックアウトされたサーバのロックを解除する場合、-u パラメータと -p パラメータで指定するアカウント名は、change password 権限を持つログイン名でなければなりません。 そのユーザが新しいパスワードでサーバにログインしたら、まず、自分のパスワードを新しいパスワードにリセットする必要があります。その後、他のログインのパスワードをリセットできます。

細密なパーミッションが有効でない場合、so_role を持つログインアカウントのリストを生成するには、-A *system_role* または --role-logins パラメータを使用します。 次に例を示します。

```
$SYBASE/$SYBASE_ASE/bin/dataserver -d master.dat -s server_name-A
sso_role
```

細密なパーミッションが有効な場合、change password 権限を持つログインアカウントのリストを生成するには、-n *system_privilege* または --permission-logins パラメータを使用します。 次に例を示します。

```
$SYBASE/$SYBASE_ASE/bin/dataserver -d master.dat -s server_name-n
"change password"
```

一般的な使用シナリオ

一般的な使用シナリオでパーミッションを設定する方法の例を示します。

シナリオ 1: アプリケーションサーバユーザのパーミッション
次の例では、すべてのユーザテーブルでの DML コマンドの実行、および任意のストアドプロシージャおよびユーザ定義関数の実行をユーザに許可します。

すべてのユーザテーブルでの DML コマンドの実行、およびデータベース db1 で定義されているストアドプロシージャおよびユーザ定義関数の実行をユーザ app_user に許可するには、db1 で manage database permissions 権限を持つユーザが次のコマンドを発行します。

```
use db1
grant insert any table to app_user
grant select any table to app_user
grant update any table to app_user
grant truncate any table to app_user
grant delete any table to app_user
grant identity_insert any table to app_user
grant identity_update any table to app_user
grant execute any procedure to app_user
grant execute any function to app_user
```

シナリオ 2: データベースアクセスマネージャのパーミッション
次の例では、ユーザ、グループ、エイリアス、およびすべてのオブジェクトに対するパーミッションの管理をユーザに許可します。

データベース db2 のユーザ、グループ、エイリアス、およびすべてのオブジェクトに対するパーミッションの管理をユーザ Joe に許可するには、db2 で manage database permissions 権限を持つユーザが次のコマンドを発行します。

```
use db2
grant manage any object permission to joe
grant manage any user to joe
```

シナリオ 3: データベースバックアップマネージャのパーミッション
次の例では、ユーザにすべてのユーザデータベースでの **dump database**、**dump transaction**、**checkpoint**、**quiesce database** の実行を許可します。

マスタデータベースのユーザで、他のユーザデータベースのユーザではないユーザ Mike に、すべてのユーザデータベースでの **dump database**、**dump transaction**、**checkpoint**、**quiesce database** の実行を許可するには、manage server permissions 権限を持つユーザが次のコマンドを発行します。

```
use master
grant dump any database to mike
grant use any database to mike
grant checkpoint any database to mike
grant quiesce any database to mike
```

シナリオ 4: ヘルプデスクオペレータのパーミッション
次の例では、ヘルプデスクオペレータにパスワードのリセットまたはプロセスの強制終了を許可します。

パスワードの問題について連絡してきたユーザのパスワードのリセットをヘルプデスクオペレータ Alice に許可するには、manage security permissions 権限を持つユーザが次のコマンドを発行します。

```
use master
grant change password to alice
```

システムリソースを使用している暴走中のユーザプロセスの強制終了を Alice に許可するには、manage server permissions 権限を持つユーザが次のコマンドを発行します。

```
use master
grant kill any process to alice
```

シナリオ 5: セキュリティ監査官のパーミッション
次の例では、情報セキュリティ部門の従業員に、SAP ASE 監査サブシステムからの日次レポートの生成を許可します。

SAP ASE 監査サブシステムからの日次レポートの生成を情報セキュリティ部門の従業員 Jane に許可するには、manage auditing 権限を持つユーザが次のコマンドを発行します。

```
use sybsecurity
sp_adduser jane
grant select any audit table to jane
```

システムテーブル master.dbo.sysprotects

データベースに対する own database などのサーバワイドの権限が付与されている場合、そのパーミッションは master.dbo.sysprotects に記録され、データベース ID (dbid) が ID フィールドに保存されます。

付与可能なパーミッションの値はすべて、master.dbo.spt_values に保存されます。 spt_values 内のカラムは、次のようにオーバロードされます。

表 26： master.dbo.spt_values

パラメータ	値
name	パーミッション名の文字列。
number	パーミッションの内部トークン番号 (sysprotects.action)。
type	パーミッションエントリの識別子。

パラメータ	値
	T - 細密なパーミッションオプションがオンであるかオフであるかに関係なく、15.7 ESD #1 以前のリリースでパーミッションが使用可能および付与可能であり、15.7 ESD #2 以降でも引き続き使用可能および付与可能である。
	GP - 細密なパーミッションオプションがオンの場合にのみ、パーミッションが使用可能および付与可能である。
ansi_w	パーミッションの実行範囲。
	1 - サーバワイド
	2 - データベースワイド
	4 - オブジェクト
low	パーミッションが最初に導入された SAP ASE のバージョン。
	0 - 15.7 ESD #2 より前のバージョンで導入。
	157002 - 15.7 ESD #2 で導入。
high	細密なパーミッションオプションがオンの場合、現在のパーミッションを管理するパーミッションのトークン番号。
	117 - `manage server permissions`
	114 - `manage security permissions`
	106 - `manage database permissions`
	97 - `manage any object permission`
msgnum	パーミッショントークンのビットセット。 • **0x00000001 サーバワイドの権限** • **0x00000002 データベースワイドの権限** • **0x00000004 オブジェクトパーミッション** • **0x00000008 権限を付与する際に、データベース名が必要である** • **0x00000010 権限は "manage server permissions" によって管理される** • **0x00000020 権限は "manage security permissions" によって管理される** • **0x00000040 権限は "manage database permissions" によって管理される** • **0x00000080 カラムレベルのパーミッション** • **0x00000100 パーミッションはシステムテーブルオブジェクトに適用される** • **0x00000200 パーミッションはビューオブジェクトに適用される** • **0x00000400 パーミッションはユーザテーブルオブジェクトに適用される** • **0x00000800 パーミッションはプロシージャオブジェクトに適用される**

データベースユーザ usedb_user アカウント

各データベースには、データベースユーザアカウント usedb_user がデフォルトで追加されます。 細密なパーミッションが有効な場合、データベースにアクセスするユーザにそのデータベースに対する use database 権限が付与されていれば、そのユーザはユーザ名 usedb_user を使用します。

ユーザには次の条件が適用されます。

- データベースに対する own database 権限を持っていない
- データベースの有効なユーザとしての ID を保持していない
- データベースの別の有効なユーザにエイリアスが設定されていない

ユーザアカウント usedb_user には次のルールが適用されます。

- usedb_user がデータベースで実行できるのは、public に許可されている操作のみです。 他の操作を実行するには、対応する権限を取得する必要があります。 データベースの有効なユーザでない場合は、役割を通じてデータベースにおける権限を取得できます。
- オブジェクトを作成するには、データベースの有効なユーザでなければなりません。

usedb_user としてのデータベースへのアクセスをユーザに許可すると、データベースの有効なユーザとして追加されていなくても、サーバワイドの権限をデータベースで実行できるようになります。

たとえば、Bob は master データベースの有効なユーザですが、db1 データベースについては有効なユーザではなく、guest ユーザアカウントを持っていないとします。 Bob は、master データベースで manage security permissions 権限を持っています。

db1 での manage security permissions の実行を Bob に許可するには、manage server permissions 権限を持つユーザが次のコマンドを発行します。

```
grant use database on db1 to bob
```

これで、Bob はデータベース db1 にアクセスし、次のコマンドを発行して、manage any encryption key 権限をユーザ Alice に付与することができます。

```
use db1
grant manage any encryption key to alice
```

SAP ASE は、manage any encryption key の付与者を usedb_user のユーザ ID として sysprotects に記録します。

付与可能なシステム権限

付与可能なサーバワイドおよびデータベースワイドの権限を次に示します。

`enable granular permissions` が無効な場合、アスタリスク (*) が付いた権限の付与または取り消しを行うことができます。

付与可能なすべての権限およびパーミッションのアルファベット順のリストについては、『リファレンスマニュアル: コマンド』の **grant** コマンドの表 1-22 を参照してください。

注意： ある権限を所有すると、さらに細密な別の権限を所有することになる場合があります。 たとえば、`select any table` 権限を持つユーザは、すべてのユーザテーブルに対する `select` パーミッションを持っていることを意味します。 暗黙的な関係を持つ権限ペアの完全なリストについては、『リファレンスマニュアル: コマンド』の **grant** コマンドの表 1-22 を参照してください。

表 27 : サーバワイドの権限

権限名	権限で許可される操作
権限管理	
`manage security permissions`	セキュリティ権限の付与と取り消し
`manage server permissions`	セキュリティに関連しない、セキュリティ以外のサーバワイドの権限の付与と取り消し
監査管理	
`manage auditing`	`sybsecurity` のデータベース所有者として操作を実施 次のコマンドの実行 • **create database sybsecurity** • **alter database** sybsecurity • **drop database sybsecurity** • **set switch** (**7601**) • **truncate table** (監査テーブル) 次のシステムストアドプロシージャの実行 • **sp_addaudittable** • **sp_audit** • **sp_displayaudit**

権限名	権限で許可される操作
ログインおよび役割管理	
allow exceptional login	次の場合にサーバへのログインを許可 • サーバの最大接続制限を超過した場合 • マスタデータベースがリストアモードになっている場合 • サーバが停止している場合 • リカバリが進行中の場合 (サーバーの再起動中)
change password	次のコマンドの実行 • **alter login ... change password** 次のシステムストアドプロシージャの実行 • **sp_password** • **sp_locklogin** (ユーザがログイン失敗の最大回数を超えたためにロックされているログインアカウントのロック解除)
manage any login	次のコマンドの実行 • **create login** • **alter login** • **drop login** 次のシステムプロシージャの実行 • **sp_addlogin** • **sp_autoconnect** • **sp_defaultdb** • **sp_displaylogin** • **sp_droplogin** • **sp_grantlogin** (Windows のみ) • **sp_helpmaplogin** • **sp_logininfo** (Windows のみ) • **sp_locklogin** • **sp_maplogin** • **sp_modifylogin** • **sp_revokelogin** (Windows のみ) 次の関数の実行 • **valid_user**

権限名	権限で許可される操作
`manage any login profile`	次のコマンドの実行 • **create login profile** • **alter login profile** • **drop login profile** 次のシステムプロシージャの実行 • **sp_securityprofile**
`manage any remote login`	次のシステムプロシージャの実行 • **sp_addexternlogin** • **sp_addremotelogin** • **sp_dropexternlogin** • **sp_dropremotelogin** • **sp_dropserver** • **sp_remoteoption**
`manage roles`	次のコマンドの実行 • **create role** • **alter role** • **drop role** • **grant role** • **revoke role** 次のシステムプロシージャの実行 • **sp_displayroles** • **sp_grantlogin** • **sp_logininfo** • **sp_revokelogin** • **sp_role**
データベース管理	
`checkpoint`	指定されたデータベースに対する **checkpoint** コマンドの実行
`checkpoint any database`	任意のデータベースに対する **checkpoint** コマンドの実行
`create database*`	**create database** コマンドの実行

権限名	権限で許可される操作
dump any database	任意のデータベースに対する次のコマンドの実行 • **dump database** • **dump transaction**
dump database	指定されたデータベースに対する次のコマンドの実行 • **dump database** • **dump transaction**
load any database	任意のデータベースに対する次のコマンドの実行 • **load database** • **load transaction**
load database	指定されたデータベースに対する次のコマンドの実行 • **load database** • **load transaction**

権限名	権限で許可される操作
`manage any database`	任意のデータベースでのデータベース管理操作の実施 次のコマンドの実行 • **install jar** • **remove jar class** 次のシステムプロシージャの実行 • **sp_addobjectdef** • **sp_addsegment** • **sp_addthreshold** • **sp_checksource** • **sp_dbextend 'simulate'** • **sp_dbextend 'execute'** • **sp_dbextend 'clear','threshold'** • **sp_dbextend 'check'** • **sp_dbextend 'set','threshold'** • **sp_dbextend 'modify', 'database'** • **sp_dbextend 'set','database'** • **sp_dbextend 'set','threshold'** • **sp_dropobjectdef** • **sp_dropsegment** • **sp_dropthreshold** • **sp_droptype** • **sp_extendsegment** • **sp_hidetext** • **sp_modifthreshold** • **sp_placeobject** • **sp_procxmode** • **sp_rename** • **sp_rebuild_text** • **sp_spaceusage** (一部のパラメータ)

権限名	権限で許可される操作
manage any database (続き)	次の **dbcc** コマンドの実行 • **dbcc dbrepair(remap)** • **dbcc dbrepair(newthreshold)** • **dbcc dbrepair(findstranded)** • **dbcc dbrepair(fixlogfreespace)** • **dbr_remap** • **dbcc dbrepair(ltmignore)** • **dbcc dbrepair(upd_usg)** • **dbcc dbrepair(auinit)** • **dbcc dbrepair(dmap_unlock)** • **dbcc rebuild_text** • **dbcc refreshids (placeobject)** • **dbcc refreshpdes** • **dbcc update_tmode** • **dbcc upgrade_obj** 次の関数の実行 • **derived_stat** • **identity_burn_max**
mount any database	任意のデータベースに対する **mount database** コマンドの実行
online any database	任意のデータベースに対する **online database** コマンドの実行
online database	指定されたデータベースに対する **online database** コマンドの実行
own any database	指定されたデータベースのデータベース所有者として操作を実施 この権限で実行できる操作のリストについては、「own database」を参照

権限名	権限で許可される操作
own database	sybsecurity を除く任意のデータベースのデータベース所有者として操作を実施 次のコマンドの実行 • **alter database** • drop database • **grant** (デフォルト) • **revoke** (デフォルト) • **checkpoint** • **dump database** • **dump transaction** • **load database** • **load transaction** • **online database** • **use database**

権限名	権限で許可される操作
`own database` (続き)	次のシステムプロシージャの実行 • **sp_addmessage** • **sp_altermessage** • **sp_dboption** • **sp_dbremap** • **sp_dropmessage** • **sp_fixindex** • **sp_forceonline_db** • **sp_forceonline_page** • **sp_helptext** • **sp_logdevice** • **sp_post_xpload** • **sp_renamedb** • **sp_setsuspect_granularity** • **sp_tempdb_markdrop** • **sp_version** • **xp_cmdshell** • **xp_deletemail** • **xp_enumgroups** • **xp_findnextmsg** • **xp_logevent** • **xp_readmail** • **xp_sendmail** • **xp_startmail** • **xp_stopmail** 次の dbcc コマンドの実行 • **dbcc addtempdb** • **dbcc dbrepair** • **dbcc reindex**
`quiesce any database`	任意のデータベースに対する **quiesce database** コマンドの実行
`unmount any database`	任意のデータベースに対する **unmount database** コマンドの実行
サーバ管理	

権限名	権限で許可される操作
`manage any thread pool`	次のコマンドの実行 • **create thread pool** • **alter thread pool** • **drop thread pool**
`manage cluster`	クラスタ関連の設定および操作の管理 **shutdown cluster** コマンドの実行 (`shutdown` 権限が必要) 次のシステムプロシージャの実行 • **sp_addserver** • **sp_cluster** • **sp_clusterlockusage** • **sp_dropserver** • **sp_serveroption** • **sp_tempdb_markdrop** 次の **dbcc** コマンドの実行 • **dbcc quorum** • **dbcc set_scope_in_cluster**
`manage disk`	次のコマンドの実行 • **disk init** • **disk refit** • **disk reinit** • **disk mirror** • **disk unmirror** • **disk remirror** 次のシステムプロシージャの実行 • **sp_addumpdevice** • **sp_diskdefault** • **sp_deviceattr** • **sp_dropdevice** • **sp_refit_admin** • **sp_dbextend** (一部のオプションに必要)

権限名	権限で許可される操作
`manage security configuration`	セキュリティ関連の設定の有効化または無効化 次のシステムプロシージャの実行 • **sp_configure** (セキュリティ関連の設定オプションの設定) • **sp_encryption** • **sp_logintrigger** • **sp_ldapadmin** • **sp_passwordpolicy** • **sp_ssladmin**

権限名	権限で許可される操作
manage server	サーバ管理操作の管理 次のシステムプロシージャの実行 • **sp_addlanguage** • **sp_addserver** (現在のセキュリティシステム担当者) • **sp_clearstats** • **sp_countmetadata** • **sp_dbrecovery_order** • **sp_displaylogin** • **sp_displayroles** • **sp_droplanguage** • **sp_dropserver** (現在のセキュリティシステム担当者) • **sp_engine** • **sp_errorlog** • **sp_extengine** • **sp_helpapptrace** • **sp_metrics** • **sp_monitorconfig** • **sp_object_stats** • **sp_processmail** (Windows のみ) • **sp_reportstats** • **sp_serveroption** • **sp_setlangalias** • **sp_tempdb** 次の **dbcc** コマンドの実行 • **dbcc complete_xact** • **dbcc engine** • **dbcc forget_xact** • **dbcc traceflags** 次の関数の実行 • **pssinfo** • **valid_user**

権限名	権限で許可される操作
manage server configuration	セキュリティに関連しないサーバ設定の有効化または無効化 次のシステムプロシージャの実行 • **sp_configure** (セキュリティ関連の設定オプションの設定) • **sp_displaylevel** • **sp_jreconfig** • **sp_lmconfig** • **sp_pciconfig**
shutdown	以下の停止 • サーバ • クラスタ (manage cluster 権限も必要) • インスタンス • Backup Server **shutdown** コマンドの実行
dbcc	
dbcc checkalloc any database	任意のデータベースでの **dbcc checkalloc** の実行
dbcc checkcatalog any database	任意のデータベースでの **dbcc checkcatalog** の実行
dbcc checkdb any database	任意のデータベースでの **dbcc checkdb** の実行
dbcc check index any database	任意のデータベースでの **dbcc checkindex** の実行
dbcc checkstorage any database	任意のデータベースでの **dbcc checkstorage** の実行
dbcc checktable any database	任意のデータベースでの **dbcc checktable** の実行
dbcc checkverify any database	任意のデータベースでの **dbcc checkverify** の実行

権限名	権限で許可される操作
dbcc fix_text any database	任意のデータベースでの **dbcc fix_text** の実行
dbcc indexalloc any database	任意のデータベースでの **dbcc indexalloc** の実行
dbcc reindex any database	任意のデータベースでの **dbcc reindex** の実行
dbcc tablealloc any database	任意のデータベースでの **dbcc tablealloc** の実行
dbcc textalloc any database	任意のデータベースでの **dbcc textalloc** の実行
dbcc tune	**dbcc tune** の実行
アプリケーション管理	
manage any execution class	次のシステムプロシージャの実行 • **sp_addengine** • **sp_addexeclass** • **sp_bindexeclass** • **sp_clearpsexe** • **sp_dropengine** • **sp_dropexeclass** • **sp_setpsexe** • **sp_unbindexeclass**
manage any ESP	次のシステムプロシージャの実行 • **sp_addextendedproc** • **sp_dropextendedproc** • **sp_freedll** • **sp_helpextendedproc**

権限名	権限で許可される操作
manage data cache	次のシステムプロシージャの実行 • **sp_bindcache** • **sp_cacheconfig** • **sp_cachestrategy** • **sp_logiosize** • **sp_poolconfig** • **sp_unbindcache** • **sp_unbindcache_all**
manage dump configuration	Backup Server のダンプ設定の管理 次のコマンドの実行 • **dump configuration** 次のシステムプロシージャの実行 • **sp_configdump** • **sp_dump history**
manage lock promotion threshold	次のシステムプロシージャの実行 • **sp_dropglockpromote** • **sp_droprowlockpromote** • **sp_setpglockpromote** • **sp_setrowlockpromote**

権限名	権限で許可される操作
monitor qp performance	クエリ処理のパフォーマンスのモニタ 次のコマンドの実行 • **set switch (3604、3605)** • **set tracefile** • **set plan for plan_list** • **set option<*optimizer_show_option*> {value | on |off}** 次のシステムプロシージャの実行 • **sp_cmp_all_qplans** • **sp_cmp_qplans** • **sp_find_qplans** • **sp_flush_query_tuning** • **sp_flushmetrics** • **sp_flushstats** • **sp_metrics ('filter'、'show'、'help' の場合)** • **sp_showplan** 次の **dbcc** コマンドの実行 • **dbcc traceoff(3604、3605)** • **dbcc traceon(3604、3605)** • **dbcc nodetraceoff(3604、3605)** • **dbcc nodetraceon(3604、3605)**
manage resource limit	次のシステムプロシージャの実行 • **sp_add_resource_limit** • **sp_add_time_range** • **sp_drop_resource_limit** • **sp_drop_time_range** • **sp_help_resource_limit** • **sp_modify_resource_limit** • **sp_modify_time_range**
その他のデッドロック	
connect*	**connect** コマンドを使用したサーバへの接続
kill	権限の所有者が所有するプロセスの強制終了
kill any process	任意のユーザが所有する任意のプロセスの強制終了

権限名	権限で許可される操作
map external file	リモートサーバ上のディレクトリまたはファイルへのプロキシテーブルのマッピング
monitor server replication	複写ステータスの表示 次のシステムプロシージャの実行 • **sp_config_rep_agent** (設定値の指定なし、データベース名の指定あり/なし) • **sp_help_rep_agent** (データベース名の指定あり/なし)
set proxy	ID を別のユーザに変更するための **set proxy** の実行
set tracing*	次のコマンドの実行 • **set tracefile** (自分自身のセッション) • **set plan for *<plan_list>* on|off** • **set option *<optimizer_show_option>*on|off** 次の **dbcc** コマンドの実行 • **dbcc traceoff(3604、3605)** • **dbcc traceon(3604、3605)** • **dbcc nodetraceoff(3604、3605)** • **dbcc nodetraceon(3604、3605)**
set tracing any process	次のコマンドの実行 • **set tracefile** (任意のセッション) • **set plan for *<plan_list>* on|off** • **set option *<optimizer_show_option>* on|off** 次の **dbcc** コマンドの実行 • **dbcc traceoff(3604、3605)** • **dbcc traceon(3604、3605)** • **dbcc nodetraceoff(3604、3605)** • **dbcc nodetraceon(3604、3605)**

権限名	権限で許可される操作
`set switch`	トレースフラグの有効化または無効化 次のコマンドの実行 • **set switch** • **show switch** 次の **dbcc** コマンドの実行 • **dbcc traceon** • **dbcc traceoff** • **dbcc nodetraceon** • **dbcc nodetraceoff** 次のストアドプロシージャの実行 • **sp_dbextend 'trace'**
`show switch`	オンになっているトレースフラグの表示 **show switch** コマンドの実行
`use any database`	権限の所有者がデータベースの有効なユーザでなく、データベースに “guest” アカウントがない場合に、任意のデータベースにアクセス **use database** コマンドの実行
`use database`	権限の所有者がデータベースの有効なユーザでなく、データベースに guest アカウントがない場合に、指定されたデータベースにアクセス **use database** コマンドの実行

表 28 : データベースワイドの権限

権限名	権限で許可される操作
パーミッションの管理	
`manage any object permission`	オブジェクトパーミッションの付与と取り消し
`manage database permissions`	データベース権限の付与と取り消し
ユーザの管理	

権限名	権限で許可される操作
manage any user	次のシステムプロシージャの実行 • **sp_addalias** • **sp_addgroup** • **sp_adduser** • **sp_changegroup** • **sp_dropalias** • **sp_dropgroup** • **sp_dropuser**
ユーザの設定	
setuser	別のユーザの代行
複写管理	
manage replication	データベースの複写設定の管理 次のコマンドの実行 • **set replication** • **set repmode** • **set repthreshold** 次のシステムプロシージャの実行 • **sp_config_rep_agent** (データベース名を指定) • **sp_help_rep_agent** (データベース名を指定) • **sp_replication_path** • **sp_reptostandby** • **sp_setrepcol** • **sp_setrepdb** • **sp_setrepdbmode** • **sp_setrepdefmode** • **sp_setreplicate** • **sp_setrepproc** • **sp_setreptable** • **sp_start_rep_agent** • **sp_stop_rep_agent** 次の **dbcc** コマンドの実行 • **dbcc gettrunc** • **dbcc settrunc**
データベースの管理	

権限名	権限で許可される操作
manage database	dbo が所有するデータにアクセスせずに、データベース管理操作を実施 次のコマンドの実行 • **install jar** • **remove jar class** 次のシステムプロシージャの実行 • **sp_addobjectdef** • **sp_addsegment** • **sp_addthreshold** • **sp_checksource** • **sp_dropobjectdef** • **sp_dropsegment** • **sp_dropthreshold** • **sp_droptype** • **sp_extendsegment** • **sp_hidetext** • **sp_merge_dup_inline_default** • **sp_modifthreshold** • **sp_placeobject** • **sp_procxmode** • **sp_rebuild_text** • **sp_spaceusage** (一部のパラメータ)

権限名	権限で許可される操作
manage database (続き)	次の **dbcc** コマンドの実行 • **dbcc dbrepair(remap)** • **dbcc dbrepair(newthreshold)** • **dbcc dbrepair(findstranded)** • **dbcc dbrepair(fixlogfreespace)** • **dbcc dbrepair(ltmignor)** • **dbcc dbrepair(updusg_anchors)** • **dbcc dbrepair(auinit)** • **dbcc dbrepair(dmap_unlock)** • **dbcc rebuild_text** • **dbcc refreshids (placeobject)** • **dbcc refreshpdes** • **dbcc update_tmode** • **dbcc upgrade_obj** 次の組み込み関数の実行 • **derived_stat** • **identity_burn_max** • lct_admin • **next_identity**
クエリプランの管理	
manage abstract plans	次のシステムプロシージャの実行 • **sp_add_qpgroup** • **sp_cmp_all_qplans** • **sp_cmp_qplans** • **sp_copy_all_qplans** • **sp_drop_all_qplans** • **sp_drop_qpgroup** • **sp_drop_qplan** • **sp_export_qpgroup** • **sp_find_qplan** • **sp_help_qpgroup** • **sp_help_qplan** • **sp_import_qpgroup** • **sp_rename_qpgroup** • **sp_set_qplan**
dbcc	

権限名	権限で許可される操作
dbcc checkalloc*	そのデータベースでの dbcc checkalloc の実行
dbcc checkcatalog*	そのデータベースでの dbcc checkcatalog の実行
dbcc checkdb*	そのデータベースでの dbcc checkdb の実行
dbcc checkindex*	そのデータベースでの dbcc checkindex の実行
dbcc checkstorage*	そのデータベースでの dbcc checkstorage の実行
dbcc checktable*	そのデータベースでの dbcc checktable の実行
dbcc checkverify*	そのデータベースでの dbcc checkverify の実行
dbcc fix_text*	そのデータベースでの **dbcc fix_text** の実行
dbcc indexalloc*	そのデータベースでの **dbcc indexalloc** の実行
dbcc reindex*	そのデータベースでの **dbcc reindex** の実行
dbcc tablealloc*	そのデータベースでの **dbcc tablealloc** の実行
dbcc textalloc*	そのデータベースでの **dbcc textalloc** の実行
manage checkstorage	データベース (権限が付与されているプロシージャで指定) における **dbcc checkstorage** 関連の設定の管理 次の **dbcc** ストアドプロシージャの実行 • **sp_dbcc_deletedb** • **sp_dbcc_deletehistory** • **sp_dbcc_evaluatedb** • **sp_dbcc_exclusions** • **sp_dbcc_patch_finishtime** • **sp_dbcc_updateconfig**

権限名	権限で許可される操作
report check-storage	データベース (権限が付与されているプロシージャで指定) での **dbcc checkstorage** の結果に関するレポートを生成するための **dbcc** プロシージャの実行 次の **dbcc** ストアドプロシージャの実行 • **sp_dbcc_configreport** • **sp_dbcc_differentialreport** • **sp_dbcc_faultreport** • **sp_dbcc_fullreport** • **sp_dbcc_recommendations** • **sp_dbcc_statisticsreport** • **sp_dbcc_summaryreport**
システムカタログ	
select any audit table	sybsecurity 内の任意の監査テーブルの選択 (sybsecurity データベースのみで使用可能)
select any system catalog	現在のデータベース内の任意のシステムテーブルからのすべてのカラムの選択
truncate any audit table	sybsecurity 内の任意の監査テーブルのトランケート (sybsecurity データベースのみで使用可能)
update any security catalog	直接更新が禁止されている、次のセキュリティ関連のシステムカタログの更新、挿入、および削除 • master.dbo.syslogins • master.dbo.syssrvroles • master.dbo.sysloginroles • db.dbo.sysroles • db.dbo.sysprotects **注意：** カタログを更新するには、設定パラメータ **allow updates to system tables** を有効にしておく必要があります。
オブジェクトの管理	
alter any object owner	データベース内の任意のオブジェクトの所有権の変更 **alter ... modify owner** コマンドの実行

権限名	権限で許可される操作
`create any object`	任意のユーザが所有する次のオブジェクトの作成 • テーブル • ビュー • プロシージャ • 関数 • デフォルト • ルール • インデックス • トリガ 次のコマンドの実行 • **create table** • **create view** • **create procedure** • **create function** • **create rule** • **create default** • **create trigger** • **create index**
`drop any object`	任意のユーザが所有する次のオブジェクトの削除 • テーブル • ビュー • プロシージャ • 関数 • デフォルト • ルール • インデックス • トリガ 次のコマンドの実行 • **drop default** • **drop function** • **drop index** • **drop procedure** • **drop rule** • **drop table** • **drop trigger** • **drop view**
暗号化の管理	

権限名	権限で許可される操作
create encryption key*	データベースでの暗号化キーの作成
manage any encryption key	任意のユーザが所有するカラム暗号化キー、マスタキー、サービスキーの作成、変更、および削除 次のコマンドの実行 • **create encryption key** • **alter encryption key** • **drop encryption key** **sp_encryption** の実行
manage column encryption key	カラム暗号化キーの作成、変更、および削除
manage master key	マスタキーの作成、変更、および削除
manage service key	サービスキーの作成、変更、および削除
デフォルト	
create default*	自分自身が所有するデフォルトの作成 **create default** コマンドの実行
create any default	任意のユーザが所有するデフォルトの作成 **create default** コマンドの実行
drop any default	任意のユーザが所有するデフォルトの削除 **drop default** コマンドの実行
関数	
create function*	自分自身が所有するユーザ定義関数の作成 次のコマンドの実行 • **create function** • **create function** (SQLJ)
create any function	任意のユーザが所有する関数の作成 次のコマンドの実行 • **create function** • **create function** (SQLJ)

権限名	権限で許可される操作
`drop any function`	任意のユーザが所有する関数の削除 次のコマンドの実行 • **drop function** • **drop function** (SQLJ)
`execute any function`	任意のユーザが所有するユーザ定義関数の実行 **execute** コマンドの実行
インデックス	
`create any index`	任意のユーザが所有するテーブルのインデックスの作成 **create index** コマンドの実行
プロシージャ	
`create procedure`	自分自身が所有するプロシージャの作成 **create procedure** コマンドの実行
`create any procedure`	任意のユーザが所有するプロシージャの作成 **create procedure** コマンドの実行
`execute any procedure`	任意のユーザが所有するプロシージャの実行 **execute** コマンドの実行
`drop any procedure`	任意のユーザが所有するプロシージャの削除 **drop procedure** コマンドの実行
ルール	
`create rule*`	自分自身が所有するルールの作成 **create rule** コマンドの実行
`create any rule`	任意のユーザが所有するルールの作成 **create rule** コマンドの実行
`drop any rule`	任意のユーザが所有するルールの削除 **drop rule** コマンドの実行
テーブル	
`alter any table`	任意のユーザが所有するユーザテーブルの変更 **alter table** コマンドの実行

権限名	権限で許可される操作
create any table	任意のユーザが所有するユーザテーブルの作成 **create table** コマンドの実行
create table*	自分自身が所有するユーザテーブルの作成 **create table** コマンドの実行
decrypt any table	暗号化されたテーブルの復号化
delete any table	任意のユーザが所有するユーザテーブルのローの削除 次のコマンドの実行 • **delete table** • **lock table**
drop any table	任意のユーザが所有するユーザテーブルの削除 **drop table** コマンドの実行
identity_insert any table	任意のユーザテーブルでの identity_update の有効化または無効化 **set identity_insert** コマンドの実行
identity_update any table	任意のユーザテーブルでの identity_insert の有効化または無効化 **set identity_update** コマンドの実行
insert any table	任意のユーザが所有するユーザテーブルの挿入 次のコマンドの実行 • **insert** • **lock table**
manage any statistics	任意のユーザが所有するテーブルの統計の更新または削除 次のコマンドの実行 • **delete statistics** • **update statistics** **sp_modifystats** の実行
references any table	任意のユーザが所有するユーザテーブルの参照

権限名	権限で許可される操作
reorg any table	任意のユーザが所有するユーザテーブルの再編成 **reorg** コマンドの実行
select any table	任意のユーザが所有するユーザテーブルの選択 次のコマンドの実行 • **select** • **lock table** (共有ロックの場合) • **readtext**
transfer any table	任意のユーザが所有するユーザテーブルとの間のデータの転送 **transfer table** コマンドの実行
truncate any table	任意のユーザが所有するユーザテーブルのトランケート **truncate table** コマンドの実行
update any table	任意のユーザが所有するユーザテーブルの更新 次のコマンドの実行 • **update** • **lock table** • **writetext**
トリガ	
create trigger*	自分自身が所有するトリガの作成 **create trigger** コマンドの実行
create any trigger	任意のユーザが所有するトリガの作成 **create trigger** コマンドの実行
drop any trigger	任意のユーザが所有するトリガの削除 **drop trigger** コマンドの実行
ビュー	
create view*	自分自身が所有するビューの作成 **create view** コマンドの実行
create any view	任意のユーザが所有するビューの作成 **create view** コマンドの実行

権限名	権限で許可される操作
drop any view	任意のユーザが所有するビューの削除 **drop view** コマンドの実行

第 9 章　データの機密保持

SAP ASE セキュリティサービスは、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、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。

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。

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。

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。 パブリックキーの暗号化には、暗号化、キー交換、デジタル署名、デジタル証明書が含まれます。

暗号化
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、改ざん検出と否認防止に使用されます。 デジタル署名は、テキストメッセージからユニークかつ固定長の数値文字列を作成する算術アルゴリズムを使用して作成されます。作成された文字列は、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ている場合は、当然、デジタル署名は否認防止に対して意味をなしません。

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。

SAP ASE 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 - サーバ証明書を保有しているサーバを認証します。 通常、証明書は、信頼されたサードパーティの認証局 (CA) 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。 SAP ASE 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。 SAP ASE 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、SAP ASE 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 対応サーバは証明書を提示し、ID を証明してから、データを送信します。 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します。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の両者が 1 つの暗号スイートについて合意すると、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 を参照してください。

SAP ASE での SSL

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します)、データ転送を行う前に、暗号化された SSL セッションを開始します。
- 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ます。
- 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ます。

SAP ASE 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 SAP ASE 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。 SAP ASE 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。

SAP ASE 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。

*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="CN=my_server_common_name.mydomain.com"
query  tcp ether myhostname myport1
ssl="CN=my_server_common_name.mydomain.com"
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、SAP ASE 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 SAP ASE 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サーバの名前です。

- 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` ファイル内のサーバ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、SAP ASE 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、SAP ASE 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、ローカルのサーバ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、SAP ASE がクライアントとして機能しているときだけです。

SAP ASE 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！ SAP ASE 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。

接続タイプ

クライアントからサーバ、およびサーバからサーバへのさまざまな接続を利用できます。

SAP ASE へのクライアントログイン
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 SAP ASE へのソケット接続を確立します。 ネットワークのトランスポート層の接続コールがクライアントサイドで完了し、受け入れコールがサーバサイドで完了すると、SSL ハンドシェイクが行われます。それから、ユーザのデータが送信されます。

サーバ間のリモートプロシージャコール
SAP ASE 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すで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。

SAP ASE 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。 SAP ASE 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 SAP ASE 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 SAP ASE の他のモジュールは、Client-Library を使用して SAP ASE 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 SAP ASE クライアント接続を認証します。

SSL の有効化

SAP ASE 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。

1. SSL 機能の SAP ASE ライセンスを取得します。 ASE_ASM または ASE_SECDIRS ライセンスを使用できます。
2. 次のコマンドを実行します。
   ```
   sp_configure "enable ssl", 1
   ```
   実行値が 1 であること、およびログファイルにエラーがないことを確認します。
3. SAP ASE サーバの署名済みサーバ証明書を要求します。 「証明書の要求」 (312 ページ)を参照してください。
   a) 証明書ファイルに、暗号化され、コード化され、ファイルに追加された証明書の「プライベートキー」があることを確認します。
   b) 対象の証明書の `commonName` は interfaces ファイルの `servername` の大文字小文字と完全に一致する必要があります。 必要に応じて、SSL フィルタ付きの証明書で `commonName` を指定できます。 「サーバディレクトリエントリの作成」 (313 ページ)を参照してください。

4. 次のコマンドを実行します。
   ```
   sp_ssladmin addcert, certificate_path, password
   ```
5. 次のロケーションにサーバ CA 証明書をインストールします。 `$SYBASE/$SYBASE_ASE/certificates/<servername>.txt`
6. サーバエントリを 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」 (313 ページ)を参照してください。
7. CA 証明書をクライアントの `trusted.txt` ファイルのロケーションにコピーします。デフォルトでは `$SYBASE/config/trusted.txt` です。

注意： サードパーティの証明書を要求、認証、変換するには、`openssl` バイナリを使用します。 バイナリの使用方法の詳細については、OpenSSL の Web サイトの OpenSSL 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。 SAP ASE 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、SAP ASE 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。

証明書の要求

システムセキュリティ担当者は、次の手順で、SAP ASE 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。

openssl を使用して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。 詳細については、OpenSSL の Web サイトの OpenSSL を参照してください。

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。 SAP ASE 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。 証明書を PEM フォーマットに変換する必要があります。 PKCS#12 フォーマットの証明書をサードパーティから要求する場合、OpenSSL を使用して、証明書を SAP ASE が解析できるフォーマットに変換してください。

FIPS が設定されている場合、証明書の生成に MD5 ハッシュアルゴリズムを使用できません。 プライベートキーは PKCS #8 暗号化形式である必要があります。

SAP ASE 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. SAP ASE インストールディレクトリに証明書を格納します。

多くのサードパーティ 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、OpenSSL 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。

注意： SAP ASE 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。

サーバディレクトリエントリの作成

SAP ASE 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。 SAP ASE 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 行のフィルタとして指定されます。 次の例は、UNIX 環境でサーバに SSL を使用する場合のエントリを示しています。

```
SYBSRV1
master tcp ether myhostname  myport
ssl="CN=my_server_common_name.mydomain.com"
query  tcp ether myhostname   myport
ssl="CN=my_server_common_name.mydomain.com"
```

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。

```
[SYBSRV1]
query=nlwnsck, myhostname,myport,ssl
master=nlwnsck, myhostname,myport,ssl
```

例における SECMECH 行には、Kerberos 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 `myport` 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

SAP ASE では、SSL と証明書の管理に **sp_ssladmin** を使用します。 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"
```

絶対パス `/sybase/ASE-12_5/certificates` (Windows の場合は `x:¥sybase¥ASE-12_5¥certificates`) の認証ファイルに、ローカルサーバのエントリ *Server1.crt* が追加されます。 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、**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！ パスワードは 64 文字に制限されています。 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サーバを再起動した後、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 SAP ASE によって暗号化されて保管されます。 その後、コマンドラインからサーバを起動するときは、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 SAP ASE 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。 これによりローカル接続だけが確立されるため、パスワードがネットワーク間の接続で送信されることはありません。

注意： SAP ASE 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すでに行っていた場合、**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 ハンドシェイクの際は、暗号スイートを介してクライアントとサーバが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 対応アプリケーションで使用す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://tools.ietf.org/html/rfc5246 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も強力な暗号スイートは、SSL ベースのセッションに使用される暗号スイートです。

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。

SAP ASE 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
```

出力は次のとおりです。

```
------------------------------
TLS_RSA_WITH_AES_128_CBC_SHA

(1 row affected)
```

SSL 暗号スイートの優先度

SAP ASE の使用する暗号スイートのセットを制限します。

暗号スイートの優先度を表示および設定するためのオプションが 2 つあります。

sp_ssladmin lsciphers と **sp_ssladmin setciphers** です。 これらのオプションによって SAP ASE 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 SAP ASE 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。 SAP ASE は、暗号スイートに対して内部的に定義された一連の設定を使用します。

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 lscipher
```

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。

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、サーバ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、SAP ASE 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。

SAP ASE 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。

SAP ASE 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、サーバ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。

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。

削除された暗号スイートを使用すると SSL ハンドシェイク が失敗し、SAP ASE には接続できません。

SAP ASE の定義済み暗号スイート

SSL 暗号スイートは、暗号スイートに対して内部的に定義された一連の設定を使用します。

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
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

次の暗号スイートは、SAP ASE バージョン 15.0 以降ではサポートされません。

- 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

sp_ssladmin の例

次の例では、**sp_ssladmin** を使用して暗号スイートの優先度を表示および設定します。

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
```

出力は次のとおりです。

```
 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、暗号スイートが SAP ASE で使用されないことを示します。 他のゼロ以外の値は、SSL ハンドシェイクの間に SAP ASE 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、SAP ASE 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
```

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 SAP ASE 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 **CN=*common_name*** パラメータが含まれており、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、SAP ASE 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。

Kerberos による機密保持

SAP ASE 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
```

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 ]
```

構文の説明は次のとおりです。

- *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 文字より短い、または 30 文字より長いパスワードを指定すると、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、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、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。

残存データの削除

データベースデータのセキュリティを強化するには、残存データを削除します。

スペースを削除するデータベース操作の中には、データが常に物理的に消去されるとは限らないものがあります。 この残存データは **dbcc** ユーティリティを使用してユーザが見ることができるため、セキュリティ脅威がもたらされる可能性があります。 ユーザがこのようなデータベース操作を実行したときに、残存データを自動的にゼロ設定できます。

`master`、`sybsystemdb`、`sybsystemprocs` などのシステムデータベースや、各データベース内にあるシステムテーブルから残存データを削除することはできません。

注意： SAP ASE が残存データのゼロ設定を終了するまで、割り付け解除された領域を再利用できません。 SAP ASE が削除する必要のあるデータ量によっては、ゼロ設定に長時間を要することがあります。

残存データが発生する操作

領域の割り付け解除が行われる操作により、残存データが発生する可能性があります。

表 29 : 残存データが発生する操作

操作	説明
drop table と **drop index**	テーブルまたはインデックスを削除すると、エクステントの割り付け解除が発生する。このエクステントは直接他のオブジェクトで再利用できるため、ユーザは **dbcc** を実行して削除されたテーブルの内容を見ることがきる。 セッションレベル、データベースレベル、テーブルレベルで **erase residual data** オプションを設定すると、残存データは削除される。
alter table	**alter table encryption** などの **alter table** コマンドにより、既存データの物理コピーが削除され、暗号化形式で格納されるが、古い ASCII (クリアテキスト) データページは削除されない。このデータページは、データベースデバイスを直接見るか、 **dbcc page** コマンドを使用すると確認できる。 セッションレベル、データベースレベル、テーブルレベルで **erase residual data** オプションを設定すると、残存データは削除される。
ワークテーブルの割り付け解除	ユーザクエリにカラムでの集計が含まれる場合、ワークテーブルにカラムのデータが存在する場合がある。 このようなワークテーブルはユーザクエリを処理し、クエリ実行の最後に割り付け解除される。 ワークテーブルはテーブルを閉じるときに割り付け解除され、そのタスクによりページの割り付けが解除される。 セッション/データベースレベルで **erase residual data** オプションを設定/有効化すると、残存データを削除できる。
ソートページの割り付け解除	**sort** は、内部リストのマージソートの内部実行をマージするために、ディスクからページを割り付ける。 外部マージに使用されるエクステントはソートメタデータ構造にキャッシュされ、ソートの最終結果を書き込みために再利用される。 外部マージによりテーブルサイズより大きいエクステントが割り付けられると、キャッシュされたエクステントがソート操作の最後に割り付け解除されて、ゼロ設定になる。
ページが空になった後のページの割り付け解除	テーブルに対する delete クエリによりページの割り付け解除が行われる。つまり、ページのすべてのローが削除されると、ページが割り付け解除される。そのようなページには残存データが存在する。

操作	説明
データベースの削除または縮小	データベースの削除/縮小の操作後に、削除されたページに物理データが残る可能性があるが、このデータはデータベースユーティリティで表示できない。 この領域を再利用する唯一の方法は、**create database** または **alter database** を実行することである。実行後にページはゼロで初期化される。
reorg rebuild	データを別の領域に複写してから、テーブルのインデックスを再作成し、データを編成する。 **reorg rebuild** は次に古いデータ (複写元のデータ) を割り付け解除する。 テーブル/セッション/データベースレベルのオプションが設定されておらず、テーブルに機密情報が含まれている場合、古いデータには引き続き残存データが存在する。
reorg compact	ページのすべてローがコミットされた削除であることがわかれば、ページの割り付けを解除する。 このようなことが行われるのは、自動ガーベジコレクションのための SAP ASE メカニズムが実行される前に **reorg compact** を実行した場合である。 **reorg compact** のためにキューイングされるページが、**erase residual data** オプションがオンになっているテーブルに存在する場合、そのページの残存データは削除される。
テンポラリテーブルの削除	テンポラリテーブルは他のテーブルと類似しているため、機密データを含むテンポラリテーブルを削除すると残存データの削除が行われる。 テンポラリテーブルの場合、セッションレベルまたはテーブルレベルで残存データの削除を設定する。これらのテーブルはデータベースオプションを設定できない `tempdb` に存在するため、データベースレベルのオプションは機能しない。

残存データの削除の有効化

データを機密としてマークし、削除操作または更新操作を実行した後に残存データを消去するように SAP ASE を設定することができます。

表 30：さまざまなレベルでの残存データの削除の有効化

レベル	説明
データベースレベル	次のコマンドを実行 **erase residual data** を true に設定すると、割り付け解除が発生する操作の実行後に、SAP ASE はすべての残存データを自動的に削除する。 **注意：** **sp_dboption** を使用しても詳細レベルの制御を行うことができず、パフォーマンスに影響を及ぼす可能性があります。 可能であれば、代わりに **set** コマンドを使用して、セッションレベルで機能を有効化することを検討してください。 デフォルトでは、データベースレベルでオプションを有効にすると、作成するすべてのテーブルが残存データを削除するように設定される。ただし、このオプションを明示的にオフにする場合はこの限りではない。 **sp_dboption** を使用してデータベースレベルのオプションを設定するには、ユーザがシステム管理者またはデータベース所有者である必要がある。
セッションレベル	次のコマンドを実行 **set** を使用することで、必要に応じて残存データの削除を有効化または無効化することができる。 このオプションをセッションレベルで有効化すると、そのセッション中のすべてのページ割り付け解除から残存データが削除される。 これには、**erase residual data** オプションを明示的にオフにしたテーブルのページ割り付け解除も含まれる。 このオプションは、個別セッションのすべてのユーザが設定でき、必要なパーミッションは特にない。

レベル	説明
テーブルレベル	新しいテーブルを作成するには、次のコマンドを実行する。 `create table table_name with "erase residual data" {on | off}` デフォルトでは設定または有効化されているものはない。つまり、機能はこのテーブルでは無効になっている。 既存のテーブルの設定を変更して、残存データの削除を有効化または無効化するには、次のコマンドを実行する。 `alter table table_name set "erase residual data" {on | off}` テーブルに **erase residual data** オプションを設定すると、SAP ASE は、残存データが発生するテーブル操作 (**drop table**、**delete row** (ページ割り付け解除が行われる)、**alter table**、**drop index**) の実行後に、割り付け解除された領域をクリーンアップする。 テーブルに対するオプションを明示的にオン/オフに設定すると、データベースの設定にかかわらず、テーブルに残存データが生じる操作に対してデータ消去が有効化/無効化される。 **create table** または **alter table** を使用してテーブルレベルのオプションを設定する場合、ユーザは、そのデータベースの `sysusers` テーブルに含まれ、**create table** パーミッションを持っている必要がある。 **alter table** の場合、ユーザはこのオプションを設定できるテーブル所有者またはデータベース所有者 (**setuser** コマンドを使用してテーブル所有者に代わることができる) でなければならない。

残存データの削除例

以下に残存データの削除例を示します。

以下の例では次の 2 つのテーブルを使用します。

- `create table t1 (col1 int) with "erase residual data" on`
- `create table t2 (col1 int) with "erase residual data" off`

例 1

`t1` の場合は、**erase residual data** オプションがオンになっているため、残存データは自動的に削除されます。一方 `t2` の場合は、**erase residual data** オプションが設定されておらず、さらに `dboption`/セッションレベルオプションが設定されていないため、残存データは自動的に削除されません。

```
drop table t1
go
drop table t2
go
```

例 2

t1 と t2 の両方の残存データが削除されます。これは、セッションレベルで設定されている **erase_residual_data** が、明示的にオフに設定されているテーブルレベルのオプション **erase_residual_data** より優先されるためです。

```
set erase_residual_data on
go
drop table t1
go
drop table t2
go
set erase_residual_data off
go
```

例 3

データベースレベルで設定されているため、テーブル t1 では残存データを消去するオプションがオンであり、t1 に対して実行される **drop table** コマンドと **truncate table** コマンドは両方とも、ページのすべての残存データをクリーンアップすることになります。

一方、テーブル t2 は **erase residual data off** 句を指定して作成されたため、**erase residual data** オプションが明示的にオフに設定されています。 このため、データベースレベルで **erase residual data** オプションが true に設定されていても、残存データは削除されません。 この結果、t2 に対して **drop table** および **truncate table** を実行した後も、残存データが保持されます。

```
create database db1
go
sp_dboption db1, "erase residual data", true
go
use db1
go
create table t1 (col int)
go
insert t1 values ...
go
create table t2 (col1 int, col2 char(10)) with "erase residual data"
off
go
truncate table t1
go
drop table t1
go
truncate table t2
go
drop table t2
go
```

例 4

この例では次のようになります。

- テーブル t1 には **erase residual data off** が明示的に設定されてはいませんが、データベースレベルで設定されているため、**truncate table t1** を実行すると t1 から残存データが削除されることになります。
- **erase residual data** オプションがデータベースレベルで設定されているため、テーブル t2 には、テーブル作成時に erase residual data オプションが設定されます。 このため、**truncate table t2** を実行すると、t2 から残存データが削除されます。
- テーブル t3 には **erase residual data off** が明示的に指定されています。このため **sp_dboption** で **erase residual data** が true に設定され、SAP ASE が **truncate table t3** を実行しても、残存データは削除されません。

```
create database db1
go
use db1
go
create table t1 (col int)
go
sp_dboption db1, "erase residual data", true
go
create table t2 (col1 int, col2 char(10))
go
create table t3 (col1 int, col2 char(10)) with "erase residual data"
off
go
truncate table t1
go
truncate table t2
go
truncate table t3
go
```

例 5

この例では次のようになります。

- t1 テーブルと t2 テーブルはどちらもデフォルトでは **erase residual data** オプションが設定されていませんが、**truncate table** コマンドの実行直前にセッションレベルで **erase_residual_data** がオンになったため、t1 と t2 の両方から残存データが削除されます。
- テーブル t3 は **erase residual data** オプションが明示的にオフに設定されていますが、セッションレベルで **erase_residual_data** オプションが設定されているため、この場合も **truncate** コマンドが実行されると、残存データは削除されます。

```
create database db1
go
```

```
use db1
go
create table t1(col int)
go
create table t2 (col1 int, col2 char(10))
go
create table t3 (col1 int, col2 char(10)) with "erase residual data"
off
go
set erase_residual_data on
go
truncate table t1
go
truncate table t2
go
truncate table t3
go
```

残存データを削除するための設定の組み合わせ

SAP ASE は、さまざまなレベルの設定内容に応じて、残存データを削除します。

表 31：設定の組み合わせと残存データの削除の動作

テーブルレベル	セッションレベル	データベースレベル	クリーンアップ
オフ	オフ	オン	はい
オフ	オン	オフ	はい
オン	オフ	オフ	はい
オン	オフ	オン	はい
オン	オン	オフ	はい
オフ	オン	オン	はい
オフ、明示的に設定	オン	オフ	はい
オフ、明示的に設定	オン	オン	はい
オフ、明示的に設定	オフ	オン	いいえ

これらの設定のうち、特に次の 2 つの組み合わせに注目してください。

- 組み合わせ 1:

テーブルレベル	セッションレベル	データベースレベル	クリーンアップ
オフ、明示的に設定	オン	オフ	はい

セッションレベルのオプションがオンの場合、テーブルレベルの **erase residual data** オプションが明示的にオフに設定されていても、テーブルの割り付け解除されたページの残存データは削除されます。
このため、現在のセッションに削除する必要のある機密情報が含まれている場合に限り、セッションレベルのオプションを有効にします。これは、このプロパティを所有しないがセッションに含まれているテーブルによっても残存データが削除されるため、パフォーマンスの低下につながるためです。

- 組み合わせ 2:

テーブルレベル	セッションレベル	データベースレベル	クリーンアップ
オフ、明示的に設定	オフ	オン	いいえ

データベースレベルのオプションがオンの場合でも、テーブルの割り付け解除されたページの残存データは削除されません。これは、テーブルレベルのオプションが明示的にオフになっているためです。
このためテーブルに機密データが含まれている場合は、テーブルレベルのオプションを明示的にオフにしないでください。これは、データベースレベルのオプションがオンになっている場合でも、割り付け解除されたページに残存データが含まれるためです。

残存データの削除が有効になっているかどうかの確認

システムプロシージャを使用して、テーブルレベル、データベースレベル、セッションレベルで残存データの削除を有効にしたかどうかを確認します。

テーブルレベルでの確認
sp_help を使用して、テーブルレベルで **erase residual data** オプションが設定されているかどうかを確認します。 次に例を示します。

```
alter table t1 set "erase residual data" on
go
sp_help t1
go

Name    Owner   Object_type    Object_status
    Create_date
------ -------- -------------- --------------------------
    ---------------------------------------
t1      dbo     user table     keep first text page erase residual data
    Jul 30 2013 10:30PM
(1 row affected)
```

データベースレベルでの確認
sp_helpdb を使用して、データベースレベルで **erase residual data** オプションが設定されているかどうかを確認します。 次に例を示します。

```
use master
go
```

```
sp_dboption db1, "erase residual data", true
go
sp_helpdb db1
go

name   db_size      owner   dbid     created         durability
     lobcomplvl   inrowlen      status
------ ------------ ------- -------- -------------- ----------
     ------------ -----------   ------------------------------
db1     700.0 MB    sa        4      Jul 22, 2013    full
     0             NULL         erase residual data
(1 row affected)
```

セッションレベルでの確認

sp_show_options を使用して、セッションレベルで **erase residual data** オプションが設定されているかどうかを確認します。 次に例を示します。

```
set erase_residual_data on
go
sp_show_options
go

number     name
------     ------------------------------------
    […]
    110    erase_residual_data

(13 rows affected)
(return status = 0)
```

制限事項

一部のオブジェクトからは残存データを削除できません。

表 32：残存データ削除の制限事項

制限事項	説明
ページレベルの割り付け解除のクリーンアップ	この機能を有効にしている限り、ページの割り付け解除レベルで残存データを削除**できる**。 ただしこの機能を有効にしても、ページの割り付け解除自体のみで残存データが削除される。ページの割り付け解除が発生しないローレベルの削除を行っても、残存データは削除されない。
ログレコード	割り付け解除後に、残存データは `syslogs` から削除されず、ログレコードはクリーンアップされない。
インメモリデータベース	インメモリデータベースに含まれるページから残存データを削除できない。
プロキシテーブル	プロキシテーブルから残存データを削除できない。

制限事項	説明
システムデータベース	master、system tempdb、sybsystemprocs などのシステムデータベースから残存データを削除できない。 また、systemtempdb に作成されたテーブルを含めた割り付け解除操作を実行したときに、残存データはデータベースレベルで削除されない。
ワークテーブルとソートページ	何らかの操作でワークテーブルまたはソートページが作成された場合、残存データの削除をデータベースレベルまたはセッションレベルで設定すると、そのページは割り付け解除される。
データベースの削除	**erase residual data** オプションがオンに設定された高い持続性を持つデータベースでデータ消失が発生することがある。つまり何らかの理由でデータベースの削除操作が停止されるかロールバックされると、データリカバリを完全に保証することはできない。

第 10 章　監査

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。 SAP ASE 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

注意： この章で説明する操作のパーミッション要件は、細密なパーミッションが無効であることを前提としています。 細密なパーミッションが有効な場合は、操作が異なることがあります。

参照：

- 第 6 章、「ユーザパーミッションの管理」 (167 ページ)

監査システム

監査システムは、sybsecurity データベース、監査キュー、設定パラメータ、システムプロシージャから構成されています。

- sybsecurity データベース - グローバル監査オプションと監査証跡を含みます。
- メモリ内の監査キュー - 監査証跡に書き込まれる前の監査レコードが送信されます。
- 設定パラメータ - 監査を管理するために使用されます。
- システムプロシージャ - 監査を管理するために使用されます。

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 SAP ASE 全体に影響します。

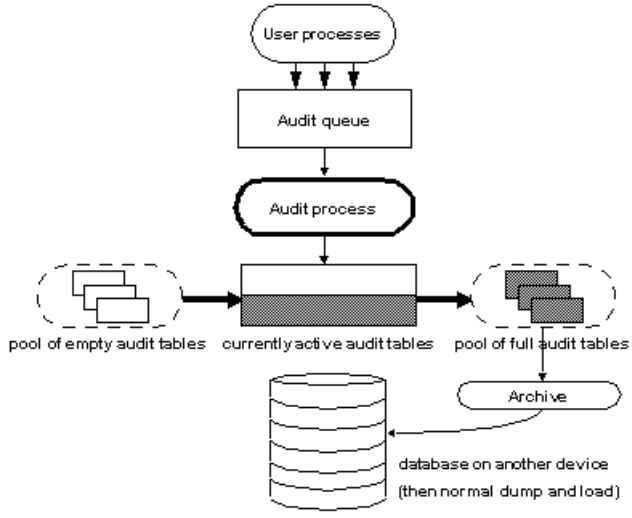
監査証跡

SAP ASE 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。 SAP ASE 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！ SAP 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。



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「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。

参照：
- 単一テーブル監査 (383 ページ)
- 監査証跡の管理 (373 ページ)

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。

監査システムの管理

監査システムを管理するためのタスクを次に示します。

1. 監査のインストール - 監査テーブル数を設定します。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ます。
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します。 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します。 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します。
3. `sybsecurity` データベースでのトランザクションログ管理の設定 - `sybsecurity` データベース内の `syslogs` トランザクションログの処理方法を決定します。この作業には、`trunc log on chkpt` データベースオプションの設定値の決定と、`trunc log on chkpt` がオフのときの `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の設定が含まれます。
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します。
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します。 SAP ASE 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します。
6. 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、**sp_audit restart** を使用して監査を再起動します。

監査システムのインストール

Adaptive Server に初めて監査をインストールする場合、2 つの方法があります。

`auditinit` ユーティリティを使用するか、インストールスクリプトを使用して、監査システムをインストールできます。 ユーティリティとインストールスクリプトは、次の場所にあります。

- UNIX:
  - `%SYBASE%/ASE-16_0/bin/auditinit`
  - `%SYBASE%/ASE-16_0/scripts/installsecurity`
- Windows:
  - `%SYBASE%¥ASE-16_0¥install¥auditinit.exe`
  - `%SYBASE%¥ASE-16_0¥scripts¥instsecu`

監査デバイスに関するインストール前の推奨事項

インストール前の推奨事項を実行してから、監査をインストールしてください。

- `sybsecurity`、`syslogs`、`sysaudits` の各テーブルデバイス用に用意するデバイスのロケーションを決定する。この情報はあとで必要になります。
- システムで必要最低限の数の監査デバイスを設定する。少なくとも 3 つのデバイスを設定する必要があります。**sp_addaudittable** を使用すると、後で監査デバイスを追加できます。『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。
- 監査テーブルとデバイスを 1 対 1 の割合でインストールする。同一デバイスを共有するテーブルは、スレッショルド上限値も共有します。これらのテーブルは、同じデバイス上にあるので、そのデバイスがいっぱいになってしまうと、続けて使用することができません。
- デバイスごとに 1 つずつ監査テーブルをインストールする。これによって、監査レコードを失うことなくスムーズに実行する監査システムを設定できます。2 つの監査テーブルを使用すると、1 つがいっぱいになったら、もう 1 つの方に切り替えることができます。さらに、3 つ目の監査テーブルを使用すると、1 つのデバイスで障害が発生した場合、システムセキュリティ担当者は新しいスレッショルドプロシージャをインストールして、障害が発生したデバイスが修復されるまでそのデバイスを省略するようにデバイスのローテーションを変更できます。
- デバイスをテーブルより大きくする。監査テーブルとデバイスを 3 つしか使用しない場合、テーブルとデバイスのサイズは同じでかまいません。監査テーブルとデバイスを追加することにより、監査機能を強化できるからです。テーブルやデバイスが上限値 (6 から 8) 近くで作業する場合は、デバイスをテーブルより十分に大きくします。監査機能を強化する必要があるものの、追加できるデバイスがほとんどまたはまったくない場合、後でテーブルのサイズをデバイスのサイズに合わせて増やすことができます。

ファイルシステムデバイスを使用する場合、**dsync** 属性をオンに設定するか、そのデバイスで **directio** 属性を使用します。

installsecurity による UNIX への監査のインストール

installsecurity スクリプトを使用して、UNIX プラットフォームに SAP ASE 監査システムをインストールします。

$SYBASE/ASE-16_0/scripts ディレクトリには、監査機能をインストールするための installsecurity スクリプトがあります。

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。

1. 監査デバイスとして使用可能なデバイス番号を決定します。
2. **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
```

3. **isql** を使用して、installsecurity スクリプトを実行します。

```
cd $SYBASE/ASE-16_0/scripts
setenv DSQUERY server_name
isql -Usa -Ppassword -Sserver_name -iinstallsecurity
```

4. SAP ASE サーバをシャットダウンして再起動します。

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** 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』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。

instsecu による Windows への監査のインストール

instsecu スクリプトを使用して、Windows プラットフォームに SAP ASE 監査システムをインストールします。

%SYBASE%¥ASE-16_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 instsecu スクリプトがあります。

1. 監査デバイスとして次に使用可能なデバイス番号を決定します。
2. [コマンド プロンプト] ウィンドウを開きます。

3. ユーザ "sa" として、**isql** プログラムを起動します。

```
isql -Usa -Ppassword -Sserver_name
```

4. **isql** プロンプトで、**disk init** コマンドを使用して監査デバイスを作成します。

   監査データベースの設定例を次に示します。

```
declare @devno int
select @devno = max(low/16777216)+1 from sysdevices
disk init
name = "auditdev",
physname = "%SYBASE%¥data¥sybaud.dat",
vdevno = @devno,
size = 5120
```

   監査データベースログの設定例を次に示します。

```
declare @devno int
select @devno = max(low/16777216)+1 from sysdevices
disk init
name = "auditlogdev",
physname = "%SYBASE%¥data¥sybaudlg.dat",
vdevno = @devno,
size = 1024
```

5. 監査データベースを作成します。

```
create database sybsecurity on auditdev
log on auditlogdev
```

6. **isql** を終了します。

```
exit
```

7. `scripts` ディレクトリに移動します。

```
cd %SYBASE%¥ASE-16_0¥scripts
```

8. DSQUERY 環境変数を設定します。

```
set DSQUERY = server_name
```

9. 入力ファイルとして **instsecu** スクリプトを使用し、**isql** プログラムをユーザ “sa” として起動します。

```
isql -Usa -Ppassword -Sserver_name  -iinstsecu
```

10. SAP ASE 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。 **sp_addaudittable** を使用して、さらに監査テーブルを追加します。 システムセキュリティ担当者が監査システムプロシージャを使用して監査を有効化するまで、監査は行われません。 **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』と『リファレンスマニュアル: コマンド』を参照してください。

auditinit による監査のインストール

auditinit ユーティリティを使用して監査をインストールします。

監査システムをインストールするプロセスには、次のようなものがあります。

- サーバの設定
- 監査テーブルのデバイスの作成
- 監査データベーストランザクションログのデバイスの作成

auditinit ユーティリティは次の場所にあります。

- UNIX: `$SYBASE/ASE-16_0/bin/auditinit`
- Windows: `%SYBASE%¥ASE-16_0¥install¥auditinit.exe`

`auditinit` ユーティリティの詳細については、『ユーティリティガイド』を参照してください。

auditinit によるサーバの設定

auditinit ユーティリティを使用して、SAP ASE サーバを監査用に設定します。

監査システムをインストールするプロセスには、次のようなものがあります。

- サーバの設定
- 監査テーブルのデバイスの作成
- 監査データベーストランザクションログのデバイスの作成

1. 次のいずれかを実行します。
   - (UNIX) source コマンドを使用して `SYBASE.csh` または `SYBASE.sh` 環境変数ファイルを実行します。
   - (Windows) ディレクトリ `%SYBASE%¥ASE-16_0¥install` に移動します。
2. コマンドウィンドウで、**auditinit** と入力します。

   次のメニューが表示されます。

   ```
   AUDITINIT
   1. Release directory:/usr/u/sybase
   2. Configure a Server product
   ```
3. [Configure a Server Product] を選択します。
4. 製品リストから [Adaptive Server] を選択するか、Windows で [SQL Server] を選択します。
5. [Configure an Exsisting Sybase Server] を選択します。
6. 設定するサーバを選択します。
7. 選択したサーバに SA パスワードを入力します。

8. [Ctrl+A] と [Enter] を押して、sa アカウントの値を受け入れます。

   auditinit のメニューに従って作業を進めるときに、表示されるデフォルト値を変更できます。 各メニューを終了するときは、[Ctrl+A] を押してデフォルト値や変更した値を受け入れて、次のメニューに移ります。
9. [Sybase Server Configuration] 画面で [Configure Auditing] を選択します。

   次のメニューが表示されます。

```
CONFIGURE AUDITING
1. Configure auditing:  no
2. Add a device for audit table(s)
3. Add a device for the audit database transaction log
4. Delete a device entry
5. Change a device entry

List of devices for the audit tables:
Logical name    Physical name    Segment name    Table name    Size

Device for the audit database transaction log:
Logical name    Physical name    Segment name    Table
name    Size
```

10. [Configure Auditing] 画面で [Configure Auditing] を選択します。

    auditint によって [Configure Auditing] メニューが再表示され、[Configure Auditing] に “yes” と表示されます。

次のステップ

監査のインストールを続行するには、監査テーブルのデバイスを追加してから、監査データベースのトランザクションログのデバイスを追加する必要があります。

参照：

- auditinit による監査テーブルのデバイスの作成 (343 ページ)
- auditinit による監査トランザクションログのデバイスの作成 (345 ページ)

auditinit による監査テーブルのデバイスの作成

auditinit を使用して監査テーブルのデバイスを作成します。

前提条件

監査テーブルのデバイスを追加する前に、**auditinit** ユーティリティを開始し、サーバを設定する必要があります。「auditinit による監査の設定」を参照してください。

手順

1. 設定するサーバを選択すると、次の `auditinit` メニューが表示されます。

```
CONFIGURE AUDITING
1. Configure auditing:  yes
2. Add a device for audit table(s)
3. Add a device for the audit database transaction log
4. Delete a device entry
5. Change a device entry

List of devices for the audit tables:
Logical name    Physical name    Segment name    Table name    Size

Device for the audit database transaction log:
Logical name    Physical name    Segment name    Table
name    Size
```

2. [Configure Auditing] 画面で [Add a Device for Audit Table(s)] を選択します。

   auditinit によって、次のメニューが表示されます。

```
ADD/CHANGE A NEW DEVICE FOR AUDITING
1. sybsecurity physical device name:
2. Logical name of the device:
3. Size of the device (Meg):
4. Device size for auditing:
```

3. [sybsecurity physical device name] を選択します。
4. 物理デバイス (ファイルシステムまたはローパーティション) のフルパスを入力します。

```
Enter the physical name of the device to use for the audit database
(default is " "):

/dev/path_to_partition
```

5. [Return] キーを押して警告を確認します。

   auditinit によって [Add/Change a New Device for Auditing] メニューが再表示され、デバイスの物理名が表示されます。

```
ADD/CHANGE A NEW DEVICE FOR AUDITING
1. sybsecurity physical device name:/secret1/sybase_dr/install/
aud1.dat
2. Logical name of the device:
3. Size of the device:
4. Device size for auditing:
```

6. このメニューの残りの項目に進みます。

   注意： [Size of the Device] の値は、[Device Size for Auditing] の値以上になるようにしてください。 [Device Size for Auditing] は、デバイスのサイズと同じにします。 SAP の監査ガイドラインに従う場合は、[Device Size for Auditing] に表示された値を変更する必要はありません。

7. [Ctrl+A] を押して、設定を受け入れます。 **auditinit** は [Configure Auditing] メニューに戻り、作成したデバイスを表示します。

```
CONFIGURE AUDITING
 1.  Configure auditing:yes
 2.  Add a device for audit table(s)
```

```
 3.  Add a device for the audit database transaction log
 4.  Delete a device entry
 5.  Change a device entry
List of devices for the audit tables:
Logical name    Physical name    Segment name
Table name    Size

6.Audit_01' secret1/sybase_dr/install/aud1.dat’ sysaudits_01 5
```

8. 複数の監査デバイスを追加するには、手順 1 ～ 6 を繰り返します。 追加できるデバイスは最高 8 つまでです。 3 つ以上の監査テーブルを追加することをおすすめします。
   デバイスを追加すると、**auditinit** は [Configure Auditing] メニューに戻り、作成したデバイスをすべて表示します。

```
CONFIGURE AUDITING
1. Configure auditing:  yes
2. Add a device for audit table(s)
3. Add a device for the audit database transaction log
4. Delete a device entry
5. Change a device entry

List of devices for the audit tables:
Logical name    Physical name    Segment name         Table
name     Size

6. Audit_01'   /secret1/sybase_dr/install/
aud1.dat’ sysaudits_01   5
7. Audit_02'   /secret1/sybase_dr/install/
aud2.dat' sysaudits_02   5
```

次のステップ

監査のインストールを続行するには、監査データベースのトランザクションログのデバイスを追加する必要があります。

参照：

- auditinit による監査トランザクションログのデバイスの作成 (345 ページ)

auditinit による監査トランザクションログのデバイスの作成

auditinit を使用して、監査トランザクションログのデバイスを作成します。

前提条件

トランザクションログのデバイスを追加する前に、**auditinit** ユーティリティを開始し、サーバを設定し、監査テーブル用のデバイスを追加する必要があります。「auditinit による監査の設定」を参照してください。

手順

1. 監査テーブルのデバイスが作成されたら、次の `auditinit` メニューが表示されます。

```
CONFIGURE AUDITING
1. Configure auditing:  yes
2. Add a device for audit table(s)
3. Add a device for the audit database transaction log
4. Delete a device entry
5. Change a device entry

List of devices for the audit tables:
Logical name    Physical name    Segment name         Table
name     Size

6. Audit_01'   /secret1/sybase_dr/install/
aud1.dat' sysaudits_01   5
7. Audit_02'   /secret1/sybase_dr/install/
aud2.dat' sysaudits_02   5
```

2. [監査機能の設定] メニューから [監査データベースのトランザクションログデバイスの追加] を選択します。

   auditinit で [監査用の新しいデバイスの追加/変更] メニューが表示されます。

```
ADD/CHANGE A NEW DEVICE FOR AUDITING
1.  sybsecurity physical device name:
2.  Logical name of the device:
3.  Size of the new device (Meg):
4.  Device size for auditing:
```

3. [sybsecurity physical device name] を選択します。

   auditinit で物理名の入力が求められ、デフォルトが表示されます (使用可能な場合)。

```
Enter the physical name of the device to use for the sybsecurity
database (default is''):
 /dev/path_to_partition
```

   path_to_partition は、デバイスのローパーティションへのパスです。

4. 物理デバイスのフルパス名を入力します。
5. [Return] キーを押します。

   auditinit で [監査用の新しいデバイスの追加/変更] メニューが表示され、デバイスの物理名に選択した値が表示されます。

```
ADD/CHANGE A NEW DEVICE FOR AUDITING
1.sybsecurity physical device name:
/secret1/sybase_dr/install/auditlog.dat
2.Logical name of the device:
3.Size of the device:
4.Device size for auditing:
```

6. このメニューの残りの項目に進みます。 その際、次の点に注意してください。
   - トランザクションログのサイズを最低 2MB にすることをおすすめします。
   - **auditinit** では、[監査用の新しいデバイスの追加/変更] メニューの [デバイスのサイズ] と [監査機能用のデバイスのサイズ] の両方にサイズが表示されます。
   - [監査機能用のデバイスのサイズ] のデフォルト値は、監査タスクのログにデバイス全体を使用するという前提に基づいて、デバイスのサイズと等しくなっています。デバイスのサブセットのみを使用し、[デバイスのサイズ] の値を編集します。
7. [Ctrl+A] を押して、[監査用の新しいデバイスの追加/変更] メニューに表示される設定を受け入れます。

   auditinit が [監査機能の設定] メニューに戻り、作成したすべてのデバイスが表示されます。

```
CONFIGURE AUDITING
1. Configure auditing:  yes
2. Add a device for audit table(s)
3. Add a device for the audit database transaction log
4. Delete a device entry
5. Change a device entry

List of devices for the audit tables:
Logical name    Physical name    Segment name         Table
name     Size
```

```
6. Audit_01'   /secret1/sybase_  dr/install/
aud1.dat' sysaudits_01   5
7. Audit_02'   /secret1/sybase_  dr/install/
aud2.dat' sysaudits_02   5
8. auditlog    /secret1/.../
auditlog.dat logsegment   syslogs        2
```

8. 監査の設定を実行する準備ができたら、[Ctrl+A] を押します。
9. [Sybase サーバの設定] 画面で、もう一度 [Ctrl+A] を押します。 次のようなメッセージが表示されます。

```
Execute the Sybase Server Configuration now?
```

10. "y" (yes) を入力します。

    auditinit によって、監査のインストール作業が実行されます。インストールが正常に完了すると、次のようなメッセージが表示されます。

```
Running task: install auditing capabilities.
.....................Done
Auditing capability installed.
Task succeeded: install auditing capabilities.
Configuration completed successfully.
Press <return> to continue.
```

参照：

- auditinit によるサーバの設定 (342 ページ)
- auditinit による監査テーブルのデバイスの作成 (343 ページ)

監査の有効化と無効化

監査をインストールしても、システムセキュリティ担当者が **sq_configure** を使用して有効にしないと監査は開始されません。

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。

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、**sp_audit** を手動で再起動できます。

次のように入力します。

```
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。

監査システムの設定

sp_configure とシステムプロシージャを使用して、監査システムを設定します。

監査プロセスの管理には、次の **sp_configure** 監査パラメータを使用します。

- **auditing** - SAP ASE サーバ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** - 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、SAP ASE サーバが再起動されるまでは有効になりません。
- **suspend audit when device full** - 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** - 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** - 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** - アクティブな監査オプションを表示します。
- **sp_addauditrecord** - 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。

ソースコードコントロールシステムでの展開されたソースの管理

SAP ASE の展開に関連付けられたすべてのユーザ作成ソースファイルをサードパーティのソースコントロールシステム (SCCS、RCS、Clearcase、Perforce など) で管理することをおすすめします。

さらに、これらのファイルの展開が自動的に行われるようにすることをおすすめします。たとえば、繰り返し可能な方法で確実にファイルをインストールするための内容明細書やスクリプトを使用します。 スクリプトソース自体をソースコントロールシステム内に保持することをおすすめします。

設定の変更の監査

SAP ASE の設定に加えられた変更も監査することをおすすめします。

sp_configure による設定の変更の監査を有効にするには、次の 2 つの方法があります。

- **sp_configure** のオブジェクトで監査オプション **exec_procedure** を使用して、**sp_configure** を直接監査する。
- **sp_configure** およびその他のプロシージャによる設定の変更を含む監査オプション **security** を使用する。

設定ファイルに直接加えられた設定の変更を取得するには、オペレーティングシステムで提供されているファイルシステム監査を設定ファイルで使用します。

設定ファイルの監査を有効にする場合は、オペレーティングシステムベンダの推奨事項に従ってください。

次に例を示します。

- Linux 2.6 カーネルでは、`auditd(8)` デーモンと関連ユーティリティ `auditctl`、`ausearch`、`aureport` を使用します。
- AIX では、監査システムは `/etc/security/audit/config` で設定し、**audit start** コマンドを使用して開始します。 AIX システムについては、IBM Redbook『Accounting and Auditing on AIX 5L』を参照してください。
- Windows では、ファイルおよびフォルダの監査を有効にするには、[セキュリティ] タブでプロパティを設定します。 たとえば、ファイルまたはフォルダを右クリックし、*プロパティ* > *セキュリティ* > *詳細設定* > *監査* を選択して、ダイアログボックスの指示に従います。 Windows の詳細については、Microsoft のサポートサイトを参照してください。

SAP ASE 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

SAP ASE 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、SAP ASE 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 SAP ASE 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。

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。

参照：

- 監査の有効化と無効化 (348 ページ)

監査オプション
監査オプションおよび要件の値を指定します。

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、サーバの起動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。

表監査オプション、要件および例には、以下が示されます。

- `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 33 : 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_addaudi-trecord** 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する。 `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する。 `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 監査をオフにする。 `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 監査をオフにする。 `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ります。 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する。 `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。			

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
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する。 `sp_audit "dbaccess", "all", "project", "on"`			
dbcc (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する。 `sp_audit "dbcc", "all", "all", "on"`			
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"`			

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
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 コマンドを監査する。 `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する。 `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する。 `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する。 `sp_audit "errors", "all", "all", "on"`			
errorlog	**all**	**all**	任意	**sp_errorlog** 関数または **errorlog_admin** 関数

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
	次の例では「ログの変更」によって新しい SAP ASE エラーログファイルに移動する試みを監査する。 `sp_audit "errorlog", "all", "all", "on"`			
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する。 `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する。 `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する。 `sp_audit @option="func_dbaccess", @login_name="all", @object_name = "strategy", @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
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する。 `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する。 `sp_audit @option="grant", @login_name="all", @object_name = "planning", @setting = "on"`			
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する。 `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する。 `sp_audit "install", "all", "planning", "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
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する。 `sp_audit "load", "all", "projects_db", "fail"`			
login (グローバル)	**all**	**all**	任意	SAP ASE へのすべてのログイン
	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する。 `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	SAP ASE からのすべてのログアウト
	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する。 `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** コマンドを監査する。 `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する。 `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する。 `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する。 `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する。 `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。
	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する。 `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**

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
	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する。 `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する。 `sp_audit "setuser", "all", "projdb", "on"`			
sproc_auth	**all**	**all**	任意	システムストアドプロシージャ内で実行される権限チェックの監査。
	次の例では、**sproc_auth** を有効にする。 `sp_audit 'sproc_auth','all','all','on'`			
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する。 `sp_audit "truncate", "all", "customer", "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
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する。 `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する。 `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する。 `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
利用可能な監査オプションの例を示します。

次の例では、`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 を監査する方法を示します。 `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** オプションを使用できます。 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
```

役割固有の監査

例 1

役割変更の監査をオンにします。

```
sp_audit "alter", "all", "master", "pass"
```

例 2

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

役割削除の監査をオフにします。

```
sp_audit "drop", "all", "master", "off"
```

例 4

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

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 '******'
```

監査が有効になっている場合のストアドプロシージャ **sp_addlogin** の例は次のとおりです。

```
sp_addlogin test2,secret
go
select event,extrainfo from sybsecurity..sysaudits_01 where event =
92
go
```

監査レコードの出力は次のようになります。

```
event extrainfo

-----------------------------------------

92 ; sp_addlogin test2,******
```

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。

grant コマンドと revoke コマンドの監査

grant コマンドと **revoke** コマンドを監査する場合、監査レコードの `extrainfo` カラムの位置 2 にコマンドテキストが書き込まれます。

"/* audit comment */" のように引用符で囲まれている場合や **"[/* audit comment */]"** のように角カッコで囲まれている場合を除き、コメントはフィルタされます。余分なスペースもフィルタされます。

grant コマンドまたは **revoke** コマンドがストアドプロシージャ内から実行された場合は、付与対象者と付与されるパーミッションが代わりに書き込まれます。

grant role sso_role to user1 の監査レコードの `extrainfo` カラムの例を次に示します。

```
sa_role sso_role oper_role sybase_ts_role mon_role; grant role
sso_role to user1; ; ; ; ; sa/ase;
```

revoke role sso_role from user1 の監査レコードの `extrainfo` カラムの例を次に示します。

```
sa_role sso_role oper_role sybase_ts_role mon_role; revoke role
sso_role from user1; ; ; ; ; sa/ase;
```

参照：

- extrainfo カラムの読み込み (387 ページ)

DML 文の監査

SAP ASE では、全文 DML 監査をサポートしています。DML 監査オプション (*table_access* と *view_access* を含む) を有効にすると、機密パラメータはマスクされた状態でバラメータ名とその値が出力されます。

次のコマンドに対する全文監査情報が記録されます。

- **select**
- **insert**
- **delete**
- **update**
- **select into**

たとえば、次の監査を設定し、

```
sp_audit "update", "all", "t1", "on"
```

テーブルを更新した場合、

```
declare @val1 int
select @val1 = 1
update t1 set c1 = @val1
```

`sysaudits_01` から監査イベントを選択すると、SAP ASE は update 文の全文を表示します (**太字**で示した部分)。

```
select event, extrainfo from sysaudits_01 where event = 70
event  extrainfo
-----------------------------------------------------------------------
70    sso_role oper_role sybase_ts_role mon_role; update t1 set c1 =
@val1; ; ; @val1=1; ; ; ; ; sa/ase;
```

SAP ASE では、ストアドプロシージャ、準備文およびアドホックバッチ内で発行された DML コマンドに対する全文 DML 監査がサポートされています。 文キャッ

シュが有効になっている場合、カーソルとともに使用する DML に対して、DML の全文も出力されます。 全文 DML 監査では、動的パラメータの名前は出力されません。

監査を有効にしたテーブルから選択を行うストアドプロシージャを作成した場合、ストアドプロシージャを実行すると、監査テーブルに **select** 文の全文が含まれます (下例の**太字**で示した部分)。

```
create proc p1 @val1 int, @val2 int
as
select * from t1 where c1 = @val1
and c2 = @val2
go
sp_audit 'select', 'all', 't1', 'on'
go
exec p1 10, 20
go
select event, extrainfo from sysaudits_01
where event = 62
event  extrainfo

------------------------------------------------------------------
 62     sa_role sso_role oper_role sybase_ts_role mon_role;

select * from t1 where c1 = @val1 and c2 = @val2; ; ; @val1=10, @val2
= 20; ; sa/ase; NULL
```

select into は **select** および **insert** 監査オプションのもとで監査されます。 次に例を示します。

```
sp_audit "select", "all", "t1", "on"
go
sp_audit "insert", "all", "default table", "on"
go
select * into t2 from t1 where c1 = 1
go
use sybsecurity
go
event  extrainfo
------------------------------------------------------------------
62     sa_role sso_role oper_role sybase_ts_role mon_role; select *
into t2 from t1; ; sa/ase;
41     sa_role sso_role oper_role sybase_ts_role mon_role; select *
into t2 from t1; ; sa/ase;
```

暗号化カラムへのデータの挿入または更新に使用されるパラメータは不明瞭にされます。 たとえば、テーブル `t1` の 2 番目のカラムは暗号化されます。

```
declare @nr char(30)
declare @nr1 int
declare @nr2 char(30)
select @nr="aaa"
select @nr1=100
select @nr2="bbb"
```

```
insert t1 values(@nr, @nr1, @nr2)
go
event  extrainfo
------------------------------------------------------------------
41     sa_role sso_role oper_role sybase_ts_role mon_role; insert t1
values(@nr, @nr1, @nr2)
                ; ; ; @nr = aaa, @nr1 =  ****** , @nr2 = bbb; ; sa/
ase;
```

全文 DML 監査では、text、unitext および image データ型のパラメータ値は出力されません。 代わりに、文字列 "Text data"、"Unitext data"、および "Image data" が結果セットにそれぞれ (**太字**で) 出力されます。

```
 event  extrainfo
------------------------------------------------------------------
41     sa_role sso_role oper_role sybase_ts_role mon_role; insert
mytext values
           (10, @a); ; ; @a = Text data; ; sa/ase;
```

注意： 全文 DML 監査用の追加領域要件を提供できるように、監査ログに対する領域のアロケーションを調整することをおすすめします。

述部付きの権限が適用された DML 文の場合、**Predicates Applied: <predicate id>** が監査レコードの `extrainfo` カラムに書き込まれます。

テーブル `t1` に対して `user1` に 2 つの述部付きの権限が付与されている **select** 文の監査レコードの例を次に示します。

```
select event, extrainfo from sysaudits_01 where event = 70
event  extrainfo
------------------------------------------------------------------
70     sso_role oper_role sybase_ts_role mon_role; ,; ; ; ,; ; ; ; ;
Predicates Applied: t1_qx8WwJltv#Co, t1_eb#rxg5pnV76; ; user1/ase;
```

Login コマンドと Login Profile コマンドの監査

login コマンドと login profile コマンドを監査する場合、機密パラメータがマスキングされた状態で、テキスト全体が `extrainfo` カラムに格納されます。

次の例では、login および login profile を含め、さまざまな文の監査レコードを示します。

createlogin 文:

```
create login test1 with passwd joe default database master
go
select event,extrainfo from sybsecurity..sysaudits_01 where
event=103
go
```

```
event extrainfo

-----------------------------------------
```

```
103 sa_role sso_role oper_role sybase_ts_role; create login test1
with passwd ****** default database master; ; ; ; ; sa/ase;
```

varchar @pass の **declare** 文を使用した　**createlogin** 文:

```
declare @pass varchar(30)
select @pass = "greatSecret"
create login test3 with passwd @pass default database master
go
select event,extrainfo from sybsecurity..sysaudits_01 where
event=103
go
```

```
event extrainfo
```

```
-----------------------------------------
```

```
103 sa_role sso_role oper_role sybase_ts_role; create login test3
with passwd @pass default database master; ; ; ; ; sa/ase;
```

暗号化パスワードを使用した **createlogin** 文:

```
create login test4 with encrypted passwd
0xc00749c449a5dd4922a59b025c605c80efe26a9235710e18b4eedb31b32edae35
6d57a4d86a57388f73c default database master
go
select event,extrainfo from sybsecurity..sysaudits_01 where
event=103
go
```

```
event extrainfo
```

```
-----------------------------------------
```

```
103 sa_role sso_role oper_role sybase_ts_role; create login test4
with encrypted passwd ****** default database master; ; ; ; ; sa/ase;
```

パスワードを変更する **alterlogin** 文:

```
alter login test1 with passwd joe123 modify passwd myPass123
go
select event,extrainfo from sybsecurity..sysaudits_01 where
event=138
go
```

```
event extrainfo
```

```
-----------------------------------------
```

```
138 ; alter login test1 with passwd ****** modify passwd
******; ; ; ; ; test1/ase;
```

create login profilejoe_lp:

```
create login profile joe_lp
go
alter login profile joe_lp modify default database "sybsystemprocs"
go
drop login profile joe_lp
```

```
select event,extrainfo from sybsecurity..sysaudits_01 where event in
(137, 140, 141)
go
```

```
event extrainfo
```

```
-----------------------------------------
```

```
138 ; sa_role sso_role oper_role sybase_ts_role; create login profile
vivekk_lp; ; ; ; ; sa/ase;
140 ; sa_role sso_role oper_role sybase_ts_role; alter login profile
vivekk_lp modify default database "sybsystemprocs"; ; ; ; ; sa/ase;
141 ; sa_role sso_role oper_role sybase_ts_role; drop login profile
vivekk_lp; ; ; ; ; sa/ase;
```

参照：

- extrainfo カラムの読み込み (387 ページ)

ストアドプロシージャの監査

ストアドプロシージャに監査オプション `sproc_auth` を使用して、システムストアドプロシージャ内で実行される権限チェックの監査を有効または無効にできます。

ストアドプロシージャの実行を監査する場合、そのプロシージャが **execute as owner** または **execute as caller** で作成されていれば、監査レコードの `extrainfo` セクションの位置 2 にその情報が書き込まれます。

`extrainfo` セクションのセクション 5 には、所有者または呼出元 (どちらか該当する方) としてユーザ名が書き込まれます。

データベース所有者が所有し、ユーザ Joe が実行し、`execute as caller` で作成されたプロシージャの `extrainfo` カラムの例を次に示します。

```
; EXECUTE AS CALLER; ; ; procedure caller = joe ; ; ;
```

Billy が所有し、`execute as owner` で作成されたプロシージャの `extrainfo` カラムの例を次に示します。

```
; EXECUTE AS OWNER; ; ; procedure owner = billy ; ; ;
```

ストアドプロシージャの権限チェックの監査

監査オプション `sproc_auth` を指定すると、システムストアドプロシージャ内で実行される権限チェックの監査が有効になります。

細密なパーミッション	イベント
有効	146
無効	80

監査イベント 80 は、監査オプション security が有効になっている場合か、監査オプション sproc_auth が有効になっている場合に監査されます。 監査イベント 146 は、オプション sproc_auth が有効になっている場合に監査されます。

sproc_auth を有効にするには、次のコマンドを実行します。

```
sp_audit 'sproc_auth','all','all','on'
```

監査システムの変更

sybsecutity を別のデバイスに移動するか、監査システム内のデバイスを変更します。

sybsecurity 監査データベースの移動

sybsecurity データベースが現在 master と同じデバイス上にある場合、既存のグローバル監査設定を保存して、または保存せずに、sybsecurity を別のデバイスに移動できます。

注意： sybsecurity データベースは、master データベースとは別に、専用のデバイスに配置されている必要があります。 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。

グローバル監査設定を保存しない sybsecurity の移動

既存のグローバル監査設定を保存せずに、sybsecurity データベースを master と同じデバイスから移動できます。

注意： sybsecurity データベースは、master データベースとは別に、専用のデバイスに配置されている必要があります。 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。

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既存のレコードをアーカイブするか、監査テーブルをアーカイブして、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。

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. 使用しているプラットフォームの設定ガイドまたは「installsecurity による監査のインストール」で説明されているインストール手順を使用して、再び sybsecurity をインストールします。
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 の移動およびグローバル監査設定の保存

sybsecurity データベースを master と同じデバイスから移動し、既存のグローバル監査設定を保存できます。

注意： sybsecurity データベースは、master データベースとは別に、専用のデバイスに配置されている必要があります。 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。

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. sybsecurity 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。

auditinit によるデバイスエントリの削除

auditinit を使用してデバイスエントリを削除します。

1. (UNIX) source コマンドを使用して `SYBASE.csh` または `SYBASE.sh` 環境変数ファイルを実行します。
2. (Windows) ディレクトリ `%SYBASE%¥ASE-16_0¥install` に移動します。
3. コマンドウィンドウで、**auditinit** と入力します。

   次のメニューが表示されます。

   ```
   AUDITINIT
   1. Release directory:/usr/u/sybase
   2. Configure a Server product
   ```

4. [Configure a Server Product] を選択します。
5. 製品リストから [Adaptive Server] を選択するか、Windows で [SQL Server] を選択します。
6. [Configure an Exsisting Sybase Server] を選択します。
7. 設定するサーバを選択します。
8. 選択したサーバに SA パスワードを入力します。
9. [Ctrl+A] と [Enter] を押して、sa アカウントの値を受け入れます。

   auditinit のメニューに従って作業を進めるときに、表示されるデフォルト値を変更できます。 各メニューを終了するときは、[Ctrl+A] を押してデフォルト値や変更した値を受け入れて、次のメニューに移ります。
10. [Sybase Server Configuration] 画面で [Configure Auditing] を選択します。

    次のメニューが表示されます。

    ```
    CONFIGURE AUDITING
    1. Configure auditing:  yes
    2. Add a device for audit table(s)
    3. Add a device for the audit database transaction log
    4. Delete a device entry
    5. Change a device entry

    List of devices for the audit tables:
    Logical name    Physical name    Segment name         Table
    name     Size
    ```

11. [Configure Auditing] - [Delete a Device Entry] を選択します。
12. 削除するデバイスの番号を入力します。
13. [Return] キーを押します。

auditinit によるデバイスエントリの変更

auditinit を使用してデバイスエントリを変更します。

1. (UNIX) source コマンドを使用して `SYBASE.csh` または `SYBASE.sh` 環境変数ファイルを実行します。
2. (Windows) ディレクトリ `%SYBASE%¥ASE-16_0¥install` に移動します。
3. コマンドウィンドウで、**auditinit** と入力します。

   次のメニューが表示されます。

   ```
   AUDITINIT
   1. Release directory:/usr/u/sybase
   2. Configure a Server product
   ```

4. [Configure a Server Product] を選択します。
5. 製品リストから [Adaptive Server] を選択するか、Windows で [SQL Server] を選択します。
6. [Configure an Exsisting Sybase Server] を選択します。
7. 設定するサーバを選択します。
8. 選択したサーバに SA パスワードを入力します。
9. [Ctrl+A] と [Enter] を押して、sa アカウントの値を受け入れます。

   auditinit のメニューに従って作業を進めるときに、表示されるデフォルト値を変更できます。 各メニューを終了するときは、[Ctrl+A] を押してデフォルト値や変更した値を受け入れて、次のメニューに移ります。
10. [Sybase Server Configuration] 画面で [Configure Auditing] を選択します。

    次のメニューが表示されます。

    ```
    CONFIGURE AUDITING
    1. Configure auditing:  yes
    2. Add a device for audit table(s)
    3. Add a device for the audit database transaction log
    4. Delete a device entry
    5. Change a device entry

    List of devices for the audit tables:
    Logical name    Physical name    Segment name         Table
    name     Size
    ```

11. [Configure Auditing] - [Change a Device Entry] を選択します。
12. 変更するデバイスの番号を入力します。

    auditinit は、[Add/Change a New Device for Auditing] メニューに、選択したデバイスの情報を表示します。

    ```
    ADD/CHANGE A NEW DEVICE FOR AUDITING
    1. sybsecurity physical device name:
    /secret1/sybase_dr/install/audlog
    ```

```
2. Logical name of the device: aud.log
3. size of the new device (Meg): 5
4. Device size for auditing:5
```

13. 変更する残りのエントリをそれぞれ選択します。
14. [Ctrl + A] を押して新しいエントリを保存します。

監査証跡の管理

次の一般的な手順では、監査証跡を効率的に管理する方法を説明しています。

- 監査機能が複数のテーブルとともにインストールされていることを確認します。各テーブルは個別のデバイスに配置されます。 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。
- 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。
- 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。
- 監査証跡と短い説明を問い合わせます。
- ユーザ指定レコードを監査証跡に追加します。

複数の sysaudits テーブルを sybsecurity に作成

SAP 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。

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。

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。

注意： 以下の手順では、監査機能を複数のテーブルとともにインストールしていることを想定しています。複数のテーブルはそれぞれ個別のデバイスに配置され

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」を参照してください。

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します。
- **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します。

現在の監査テーブルの変更

current audit table 設定パラメータは、SAP ASE 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サーバに指示します。 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、新規テーブルが空白になっていない場合は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： 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。

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、**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「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、SAP ASE 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、このように手動操作なしで続きます。

監査設定パラメータ

監査機能のインストールには、**audit queue size** および **suspend audit when device full** 設定パラーメタを使用します。

次の設定パラーメタを設定します。

- **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 SAP ASE 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。

監査キューのサイズの管理

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。

監査キューのデフォルトサイズは 100 バイトです。

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「設定パラメータの設定」 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 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 SAP ASE 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。

監査証跡の問い合わせ

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」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

現在の監査テーブルにレコードを追加し、extrainfo および dbname カラムに情報を挿入する例を示します。

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 カラムにのみ情報を挿入します。

```
sp_addauditrecord @text="I am disabling auditing
briefly while we reconfigure the system",
@db_name="corporate"
```

トランザクションログのトランケート

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。 SAP ASE 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」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

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と、セグメントの空き領域がスレッショルドの量を下回ったときにプロシージャを起動できます。

sp_dboption 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、SAP ASE 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。 スレッショルドの量は、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」を参照してください。

SAP ASE 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「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。

単一テーブル監査

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。

SAP 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。

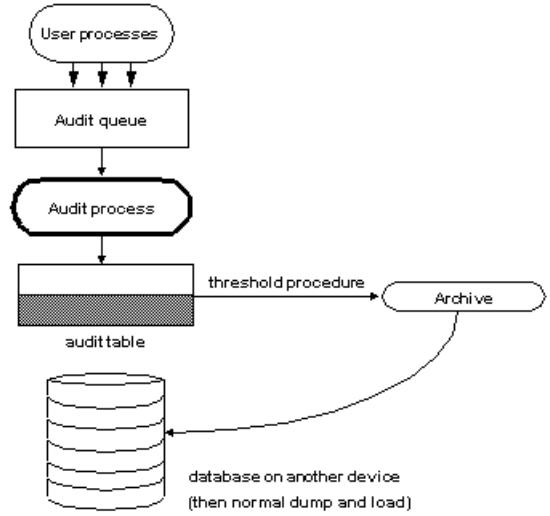
非運用システムの場合、少数の監査レコードの消失であれば、それほど問題はありません。このため、複数の監査テーブルを使用するためのディスク領域を追加できない場合や、使用できる追加デバイスがない場合は、単一テーブルを使用して監査を行います。

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。

単一の監査テーブルによる監査



単一テーブル監査のスレッショルドプロシージャ

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。 手順については、「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」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」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ったときのイベントをリストします。

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。 エラーメッセージが、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

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」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シャットダウン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。

監査テーブル

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 34 : 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。
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。 有効な値: • 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)。

カラム名	データ型	説明
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。
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 35 : 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。

位置	カテゴリ	説明
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	プロキシ情報	**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 SAP ASE 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 36 : 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)	監査を有効化するコマンド： **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)	監査を無効化するコマンド： **sp_configure auditing**	74	

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
管理者のアカウントのロック解除	監査を無効化するコマンド： **sp_configure auditing**	74	
adhoc	ユーザ定義監査レコード	1	extrainfo には、**sp_addauditrecord** の text パラメータが挿入される
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:* • ユーザ定義データ型の場合: オプションが指定されている場合は *owner.obj_namename_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*	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**。

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
	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	
bind	**sp_bindefault**	6	*その他の情報:* デフォルト名
	sp_bindmsg	7	*その他の情報:* メッセージ ID
	sp_bindrule	8	*その他の情報:* ルール名
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 or replace** コマンドの場合は **or replace**
	create procedure	11	**create or replace** コマンドの場合は **or replace**
	create rule	13	**create or replace** コマンドの場合は **or replace**
	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る。 **EK*column1*/*keyname1*[,*column2keyname2*]** **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、また、*keyname* はキーの完全修飾名。
	create trigger	12	**create or replace** コマンドの場合は **or replace**

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
	create view	16	**create or replace** コマンドの場合は **or replace**
	create index	104	*その他の情報:* インデックス名
	create function	97	**create or replace** コマンドの場合は **or replace**
	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**
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:* ディスク名
	disk mirror	23	*キーワードまたはオプション:* **disk mirror** *その他の情報:* ディスク名
	disk refit	21	*キーワードまたはオプション:* **disk refit** *その他の情報:* ディスク名
	disk reinit	22	*キーワードまたはオプション:* **disk reinit** *その他の情報:* ディスク名
	disk release	87	*キーワードまたはオプション:* **disk release** *その他の情報:* ディスク名
	disk remirror	25	*キーワードまたはオプション:* **disk remirror** *その他の情報:* ディスク名
	disk unmirror	24	*キーワードまたはオプション:* **disk unmirror** *その他の情報:* ディスク名

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 resize	100	*キーワードまたはオプション:* **disk resize** *その他の情報:* ディスク名
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: **algorithm name-bitlength/IV [random|NULL]/pad [random |NULL] user/system** 例: `AES-128/IV RANDOM/PAD NULL USER`
	alter encryption key	108	`default/not default`

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
	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る。 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る。 `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る。
	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。
	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。
	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る。 『暗号化カラムユーザーズガイド』を参照。
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る。 **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*

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
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る)。	86	
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** に続けて、完全修飾されたオブジェクト名
	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

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
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名 *その他の情報:* サーバ名 (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る。 • 役割アクティブ化の試行回数が **max failed_logins** に達したため SAP ASE によって役割がロックされる。
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

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
	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:* 設定される役割の名前
	サーバの起動	50	*その他の情報:* • `-dmasterdevicename` • `-iinterfaces file path` • `-Sservername` • `-eerrorfilename`
	sp_webservices	111	*キーワードまたはオプション:* 単一の Web サービスを展開する場合は **deploy** すべての Web サービスを展開する場合は **deploy_all**
	sp_webservices	111	*キーワードまたはオプション:* 単一の Web サービスの展開を解除する場合は **deploy** すべての Web サービスの展開を解除する場合は **deploy_all**
	サーバのシャットダウン	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 add-cert** を含むキーワード

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**、**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 SAP ASE 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 ステータスとセカンダリ URL ステータス • 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る。
	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**

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
	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:* 設定されたユーザの名前
sproc_auth	システムストアドプロシージャ内で実行される権限チェックの監査。	80、146	
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
update	テーブルの **update**	70	*キーワードまたはオプション:* • **update** • **writetext**

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
	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 37 : 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**	68	**unbind rule**
3	**alter table**	69	**unbind message**
4	**bcp in**	70	**update table**
5	予約済み	71	**update view**
6	**bind default**	72	予約済み
7	**bind message**	73	監査の有効化
8	**bind rule**	74	監査の無効化
9	**create database**	75	予約済み
10	**create table**	76	SSO が変更したパスワード
11	**create procedure**	77	予約済み

監査イベント ID	コマンド名	監査イベント ID	コマンド名
12	**create trigger**	78	予約済み
13	**create rule**	79	予約済み
14	**create default**	80	役割チェックの実行 システムストアドプロシージャ内で実行される権限チェックの監査
15	**create message**	81	**dbcc**
16	**create view**	82	config
17	**access to database**	83	**online database**
18	**delete table**	84	**setuser** コマンド
19	**delete view**	85	**create role**、**drop role**、**alter role**、**grant role**、または **revoke role**
20	**disk init**	86	組み込み関数
21	**disk refit**	87	ディスクの解放
22	**disk reinit**	88	**set SSA** コマンド
23	**disk mirror**	89	**kill** コマンドまたは **terminate** コマンド
24	**disk unmirror**	90	**connect**
25	**disk remirror**	91	reference
26	**drop database**	92	コマンドテキスト
27	**drop table**	93	**JCS install** コマンド
28	**drop procedure**	94	**JCS remove** コマンド
29	**drop trigger**	95	管理者アカウントのロック解除
30	**drop rule**	96	**quiesce database**
31	**drop default**	97	**create SQLJ** 関数
32	**drop message**	98	**drop SQLJ** 関数
33	**drop view**	99	SSL 管理
34	**dump database**	100	**disk resize**
35	**dump transaction**	101	**mount database**

監査イベント ID	コマンド名	監査イベント ID	コマンド名
36	致命的なエラー	102	**unmount database**
37	致命的ではないエラー	103	**create login**
38	ストアドプロシージャの実行	104	**create index**
39	トリガの実行	105	**drop index**
40	**grant**	106	**sp_encryption** (暗号化された列の管理)
41	**insert table**	107	**create encryption key**
42	**insert view**	108	**Alter Encryption Key as/not default**
43	**load database**	109	**drop encryption key**
44	**load transaction**	110 111	**deploy user-defined web services** **undeploy user defined web services**
45	**login**	112	ログインがロックされている
46	**logout**	113	**quiesce hold security**
47	**revoke**	114	**quiesce release**
48	**rpc in**	115	パスワード管理
49	**rpc out**	116	**create manifest file**
50	サーバの起動	117	**regenerate keypair**
51	サーバのシャットダウン	118	**alter encryptin key modify encryption**
52	予約済み	119	**alter encryption key add encryption**
53	予約済み	120	**alter encryption key drop encryption**
54	予約済み	121	**alter encryption key modify owner**
55	役割のオンとオフ	122	**alter encryption key for key recovery**
56	予約済み	123	LDAP ステータス変更
57	予約済み	124	**alter...modify owner**
58	予約済み	127	エラーログの管理
59	予約済み	136	**transfer table**
60	予約済み	136	**transfer table**

監査イベント ID	コマンド名	監査イベント ID	コマンド名
61	監査テーブルへのアクセス	137	**create login profile**
62	テーブルからの **select**	138	**alter login**
63	ビューからの **select**	139	**drop login**
64	**truncate table**	140	**alter login profile**
65	予約済み	141	**drop login profile**
66	予約済み	142	予約済み
67	**unbind default**	143	**alter thread pool**
		146	**sproc_auth**

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、SAP ASE 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。

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 SAP ASE 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 SAP ASE 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)。
- SAP ASE 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。
- SAP ASE 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 SAP ASE 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。
  - 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。
- SAP ASE 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。
- 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。
- SAP ASE 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。
- SAP ASE 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。