SYBASE®

システム管理ガイド :
第 1 巻

Adaptive Server® Enterprise

15.5

ドキュメント ID：DC36426-01-1550-01
改訂：2009 年 11 月

マニュアルの注文

マニュアルの注文を承ります。ご希望の方は、サイベース株式会社営業部または代理店までご連絡ください。マニュアルの変更は、弊社の定期的なソフトウェア・リリース時にのみ提供されます。

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

目次

はじめに

『システム管理ガイド : 第 1 巻』は、どのような特定のデータベース・アプリケーションからも独立して、Sybase® Adaptive Server® Enterprise データベースを管理および制御する方法を説明します。

対象読者

このマニュアルは、Sybase のシステム管理者およびデータベース所有者を対象としています。

このマニュアルの内容

このマニュアルは、2 つのパートで構成されています。パート 1 では、システム管理に関する基本概念について説明しています。内容は、次のとおりです。

- 「第 1 章 システム管理の概要」では、Sybase システムの構造について説明します。
- 「第 2 章 システム・データベースとオプションのデータベース」では、Adaptive Server システム・データベースの内容と機能について説明します。
- 「第 3 章 システム管理の基礎」では、新たにシステム管理者となった方が実行する必要のある重要な作業について説明します。
- 「第 4 章 Adaptive Server Plug-in for Sybase Central の概要」では、Adaptive Server を管理するためのグラフィカル・ユーザ・インタフェース、Sybase Central を起動および使用する方法について説明します。
- 「第 5 章 設定パラメータ」では、設定パラメータについて説明します。これらのパラメータを sp_configure で設定することによって、Adaptive Server のさまざまな機能を制御できます。
- 「第 6 章 ディスク・リソースについての概要」では、Adaptive Server と Backup Server™ のエラー処理、およびサーバの停止方法とユーザ・プロセスの強制終了方法について説明します。
- 「第 7 章 データベース・デバイスの初期化」では、データベース・デバイスの初期化方法およびデバイスをデフォルト・デバイス・プールに割り当てる方法について説明します。
- 「第 8 章 データベース・オプションの設定」では、データベース・オプションの設定方法について説明します。
- 「第 9 章 文字セット、ソート順、言語の設定」では、言語モジュールに含まれるファイルなどの国際化についての問題、および Adaptive Server の言語、ソート順、文字セットの設定方法について説明します。

- 「第 10 章 クライアント／サーバの文字セット変換の設定」では、異機種間環境での Adaptive Server とクライアント間の文字セット変換について説明します。
- 「第 11 章 システムの問題の診断」では、Adaptive Server と Backup Server のエラー処理、およびサーバの停止方法とユーザ・プロセスの強制終了方法について説明します。

パート 2 では、セキュリティ管理について説明しています。内容は、次のとおりです。

- 「第 12 章 セキュリティの概要」では、セキュリティの概念を紹介します。
- 「第 13 章 Adaptive Server のセキュリティ管理について」では、Adaptive Server で使用できるセキュリティ機能の概要について説明します。
- 「第 14 章 Adaptive Server のログイン、データベース・ユーザ、クライアント接続の管理」では、Adaptive Server のログイン・アカウントとデータベース・ユーザを管理する方法について説明します。
- 「第 15 章 リモート・サーバの管理」では、各 Adaptive Server のシステム管理者とシステム・セキュリティ担当者が、リモート・プロシージャ・コール (RPC) を使用できるようにするために実行する手順について説明します。
- 「第 16 章 外部認証」では、ユーザの認証とネットワークを介して転送するデータの保護を可能にする、ネットワーク・ベースのセキュリティ・サービスについて説明します。
- 「第 17 章 ユーザ・パーミッションの管理」では、ユーザ・パーミッションの使用と実装について説明します。
- 「第 18 章 監査」では、インストール環境に応じた監査の設定方法について説明します。
- 「第 19 章 データの機密保持」では、すべてのデータを保護し、機密性を保持するための Adaptive Server の設定方法について説明します。

関連マニュアル

Adaptive Server® Enterprise には次のマニュアルが用意されています。必要に応じて参照してください。

- 使用しているプラットフォームの『リリース・ノート』－ マニュアルには記載できなかった最新の情報が記載されています。

  このリリース・ノートの最新バージョン (英語版) を入手できます。製品の CD がリリースされた後で、製品またはマニュアルに関する重要な情報が追加されているかを確認するには、Sybase® Product Manuals Web サイトを使用してください。
- 使用しているプラットフォームの『インストール・ガイド』－ すべての Adaptive Server および関連する Sybase 製品のインストール、アップグレード、設定の手順について説明しています。

- 『新機能ガイド』－ Adaptive Server の新しい機能について説明しています。また、新しい機能をサポートするためのシステム変更や、既存のアプリケーションに影響を与える可能性がある変更についても説明しています。
- 『Active Messaging ユーザーズ・ガイド』－ Active Messaging を使用して、Adaptive Server Enterprise データベースでトランザクション (データ変更) を取得し、外部アプリケーションにイベントとしてリアルタイムで渡す方法について説明しています。
- 『コンポーネント統合サービス・ユーザーズ・ガイド』－ コンポーネント統合サービスを使用して、リモートの Sybase データベースおよび Sybase 以外のデータベースに接続する方法について説明しています。
- 使用しているプラットフォームの『設定ガイド』－ 特定の設定作業の手順について説明しています。
- 『用語解説』－ Adaptive Server マニュアルで使用されている技術用語について説明しています。
- 『Historical Server ユーザーズ・ガイド』－ Historical Server を使用して、Adaptive Server のパフォーマンス情報を入手する方法について説明しています。
- 『Adaptive Server Enterprise における Java』－ Adaptive Server データベースで Java クラスをデータ型、関数、ストアド・プロシージャとしてインストールして使用する方法について説明しています。
- 『Job Scheduler ユーザーズ・ガイド』－ コマンド・ラインまたはグラフィカル・ユーザ・インタフェース (GUI) を使用して、ローカルまたはリモートの Adaptive Server でジョブのインストール、設定、作成、スケジュールを行う方法について説明しています。
- 『マイグレーション技術ガイド』－ 別のバージョンの Adaptive Server にマイグレートするための方法とツールについて説明しています。
- 『Monitor Client Library プログラマーズ・ガイド』－ Adaptive Server のパフォーマンス・データにアクセスする Monitor Client Library アプリケーションの記述方法について説明しています。
- 『Monitor Server ユーザーズ・ガイド』－ Monitor Server を使用して、Adaptive Server のパフォーマンス統計を取得する方法について説明しています。
- 『モニタリング・テーブル・ダイヤグラム』－ モニタリング・テーブルと、そのエンティティの関係をポスター形式で図解しています。フル・サイズのダイアグラムは印刷版だけで参照できます。コンパクト版は PDF 形式で参照できます。
- 『パフォーマンス&チューニング・シリーズ』－ Adaptive Server で最高のパフォーマンスを実現するためのチューニング方法について説明しています。このマニュアルは以下の 7 冊に分かれています。

- 『基本』 – Adaptive Server のパフォーマンスに関する問題の理解と調査の基本について説明しています。
- 『統計的分析によるパフォーマンスの向上』 – Adaptive Server で統計情報がどのように保存され、表示されるかについて説明しています。また、set statistics コマンドを使用して、サーバの統計情報を分析する方法について説明しています。
- 『ロックと同時実行制御』 – ロック・スキームを使用してパフォーマンスを向上させる方法と、同時実行性を最小限に抑えるようにインデックスを選択する方法について説明しています。
- 『sp_sysmon による Adaptive Server の監視』 – sp_sysmon を使用してパフォーマンスをモニタリングする方法について説明しています。
- 『モニタリング・テーブル』 – Adaptive Server のモニタリング・テーブルに統計情報や診断情報を問い合わせる方法について説明しています。
- 『物理データベースのチューニング』 – データの物理的配置、データに割り付けられた領域、テンポラリ・データベースの管理方法について説明しています。
- 『クエリ処理と抽象プラン』 – オプティマイザがクエリを処理する方法と、抽象プランを使用してオプティマイザのプランの一部を変更する方法について説明しています。

- 『クイック・リファレンス・ガイド』 – コマンド、関数、システム・プロシージャ、拡張システム・プロシージャ、データ型、ユーティリティの名前と構文の包括的な一覧表を記載したポケット版 (PDF 版は通常サイズ) のマニュアルです。
- 『リファレンス・マニュアル』 – 詳細な Transact-SQL® 情報を記載しています。
  - 『ビルディング・ブロック』 – データ型、関数、グローバル変数、式、識別子とワイルドカード、予約語について説明しています。
  - 『コマンド』 – コマンドについて説明しています。
  - 『プロシージャ』 – システム・プロシージャ、カタログ・ストアド・プロシージャ、システム拡張ストアド・プロシージャ、dbcc ストアド・プロシージャについて説明しています。
  - 『テーブル』 – システム・テーブル、モニタリング・テーブル、dbcc テーブルについて説明しています。

- 『システム管理ガイド』－
  - 『第 1 巻』－ 設定パラメータ、リソースの問題、文字セット、ソート順、システムの問題の診断方法に関する説明を含め、システム管理の基本の概要について説明しています。『第 1 巻』の後半は、セキュリティ管理に関する詳細な説明です。
  - 『第 2 巻』－ 物理的なリソースの管理、デバイスのミラーリング、メモリとデータ・キャッシュの設定、マルチプロセッサ・サーバとユーザ・データベースの管理、データベースのマウントとマウント解除、セグメントの作成と使用、reorg コマンドの使用、データベース一貫性の検査方法についての手順とガイドラインを説明しています。『第 2 巻』の後半では、システムとユーザ・データベースをバックアップおよびリストアする方法について説明しています。
- 『システム・テーブル・ダイアグラム』－ システム・テーブルと、そのエンティティとの関係をポスター形式で図解しています。フル・サイズのダイアグラムは印刷版だけで参照できます。コンパクト版は PDF 形式で参照できます。
- 『Transact-SQL ユーザーズ・ガイド』－ リレーショナル・データベース言語の拡張版である Sybase の Transact-SQL について説明しています。まだ経験の浅いデータベース管理システムのユーザは、このマニュアルをガイドブックとして使用してください。pubs2 および pubs3 サンプル・データベースの詳細も説明しています。
- 『トラブルシューティング：エラー・メッセージと詳細な解決方法』－ 発生する可能性のある問題について、トラブルシューティング手順を説明しています。このマニュアルで取り上げられている問題は、Sybase 製品の保守契約を結んでいるサポート・センタに最も頻繁に寄せられるものです。
- 『クエリ処理と抽象プラン』－ Adaptive Server を使用して暗号化カラムを設定し、使用する方法について説明しています。
- 『インメモリ・データベース・ユーザーズ・ガイド』－ イン・メモリ・データベースの設定および使用方法について説明しています。
- 『Adaptive Server 分散トランザクション管理機能の使用』－ 分散トランザクション処理環境での Adaptive Server DTM 機能の設定、使用、トラブルシューティングについて説明しています。
- 『IBM® Tivoli® Storage Manager と Backup Server の使用』－ IBM Tivoli Storage Manager を設定および使用して Adaptive Server のバックアップを作成する方法について説明しています。
- 『高可用性システムにおける Sybase フェールオーバの使用』－ Sybase のフェールオーバ機能を使用して、Adaptive Server を高可用性システムのコンパニオン・サーバとして設定する方法について説明しています。

- 『Unified Agent および Agent Management Console』－ Unified Agent について説明しています。Unified Agent は、分散 Sybase リソースを管理、モニタ、制御するためのランタイム・サービスを提供します。
- 『ユーティリティ・ガイド』－ オペレーティング・システム・レベルで実行される isql および bcp などの、Adaptive Server のユーティリティ・プログラムについて説明しています。
- 『Web Services ユーザーズ・ガイド』－ Adaptive Server 用の Web サービスの設定、使用、トラブルシューティング方法について説明しています。
- 『XA インタフェース統合ガイド for CICS、Encina、TUXEDO』－ X/Open XA トランザクション・マネージャを備えた Sybase DTM XA インタフェースを使用する方法について説明しています。
- 『Adaptive Server Enterprise における XML サービス』では、データベースに XML 機能を導入する、Sybase ネイティブの XML プロセッサと Sybase Java ベースの XML のサポートについて、また XML サービスで使用できるクエリとマッピング用の関数について説明しています。

その他の情報

Sybase Getting Started CD、SyBooks™ CD、Sybase Product Manuals Web サイトを利用すると、製品について詳しく知ることができます。

- Getting Started CD には、PDF 形式のリリース・ノートとインストール・ガイド、SyBooks CD に含まれていないその他のマニュアルや更新情報が収録されています。この CD は製品のソフトウェアに同梱されています。Getting Started CD に収録されているマニュアルを参照または印刷するには、Adobe Acrobat Reader が必要です (CD 内のリンクを使用して Adobe の Web サイトから無料でダウンロードできます)。
- SyBooks CD には製品マニュアルが収録されています。この CD は製品のソフトウェアに同梱されています。Eclipse ベースの SyBooks ブラウザを使用すれば、使いやすい HTML 形式のマニュアルにアクセスできます。

  一部のマニュアルは PDF 形式で提供されています。これらのマニュアルは SyBooks CD の PDF ディレクトリに収録されています。PDF ファイルを開いたり印刷したりするには、Adobe Acrobat Reader が必要です。

  SyBooks をインストールして起動するまでの手順については、Getting Started CD の『SyBooks インストール・ガイド』、または SyBooks CD の *README.txt* ファイルを参照してください。
- Sybase Product Manuals Web サイトは、SyBooks CD のオンライン版であり、標準の Web ブラウザを使用してアクセスできます。また、製品マニュアルのほか、EBFs/Updates、Technical Documents、Case Management、Solved Cases、ニュース・グループ、Sybase Developer Network へのリンクもあります。

  Technical Library Product Manuals Web サイトにアクセスするには、Product Manuals (http://www.sybase.com/support/manuals/) にアクセスしてください。

Web 上の Sybase 製品の動作確認情報

Sybase Web サイトの技術的な資料は頻繁に更新されます。

❖ **製品認定の最新情報にアクセスする**

1 Web ブラウザで Technical Documents を指定します。(http://www.sybase.com/support/techdocs/)

2 [Certification Report] をクリックします。

3 [Certification Report] フィルタで製品、プラットフォーム、時間枠を指定して [Go] をクリックします。

4 [Certification Report] のタイトルをクリックして、レポートを表示します。

❖ **コンポーネント認定の最新情報にアクセスする**

1 Web ブラウザで Availability and Certification Reports を指定します。(http://certification.sybase.com/)

2 [Search By Base Product] で製品ファミリとベース製品を選択するか、[Search by Platform] でプラットフォームとベース製品を選択します。

3 [Search] をクリックして、入手状況と認定レポートを表示します。

❖ **Sybase Web サイト (サポート・ページを含む) の自分専用のビューを作成する**

MySybase プロファイルを設定します。MySybase は無料サービスです。このサービスを使用すると、Sybase Web ページの表示方法を自分専用にカスタマイズできます。

1 Web ブラウザで Technical Documents を指定します。(http://www.sybase.com/support/techdocs/)

2 [MySybase] をクリックし、MySybase プロファイルを作成します。

Sybase EBF とソフトウェア・メンテナンス

❖ **EBF とソフトウェア・メンテナンスの最新情報にアクセスする**

1 Web ブラウザで Sybase Support Page を指定します。(http://www.sybase.com/support)

2 [EBFs/Maintenance] を選択します。MySybase のユーザ名とパスワードを入力します。

3 製品を選択します。

4 時間枠を指定して [Go] をクリックします。EBF/Maintenance リリースの一覧が表示されます。

鍵のアイコンは、「Technical Support Contact」として登録されていないため、一部の EBF/Maintenance リリースをダウンロードする権限がないことを示しています。未登録でも、Sybase 担当者またはサポート・コンタクトから有効な情報を得ている場合は、[Edit Roles] をクリックして、「Technical Support Contact」の役割を MySybase プロファイルに追加します。

5 EBF/Maintenance レポートを表示するには [Info] アイコンをクリックします。ソフトウェアをダウンロードするには製品の説明をクリックします。

表記規則

次の項では、このマニュアルで使用されている表記について説明します。

SQL は自由な形式の言語で、1 行内のワード数や、改行の仕方に規則はありません。このマニュアルでは、読みやすくするため、例や構文を文の句ごとに改行しています。複数の部分からなり、2 行以上にわたる場合は、字下げしています。複雑なコマンドの書式には、修正された BNF (Backus Naur Form) 記法が使用されています。

表 1 に構文の規則を示します。

表 1: このマニュアルでのフォントと構文規則

要素	例
コマンド名、プロシージャ名、ユーティリティ名、その他のキーワードは sans serif フォントで表記する。	select sp_configure
データベース名とデータ型は sans serif フォントで表記する。	master データベース
ファイル名、変数、パス名は斜体で表記する。	システム管理ガイド *sql.ini* ファイル *column_name* *$SYBASE/ASE* ディレクトリ

変数 (ユーザが入力する値を表す語) がクエリまたは文の一部である場合は Courier フォントの斜体で表記する。	`select column_name`
	`from table_name`
	`where search_conditions`

カッコはコマンドの一部として入力する。	compute *row_aggregate* (*column_name*)
2 つのコロンと等号は、構文が BNF 表記で記述されていることを示す。この記号は入力しない。「〜と定義されている」ことを意味する。	`::=`
中カッコは、その中のオプションを 1 つ以上選択しなければならないことを意味する。コマンドには中カッコは入力しない。	`{cash, check, credit}`
角カッコは、オプションを選択しても省略してもよいことを意味する。コマンドには角カッコは入力しない。	`[cash | check | credit]`
中カッコまたは角カッコの中のカンマで区切られたオプションをいくつでも選択できることを意味する。複数のオプションを選択する場合には、オプションをカンマで区切る。	`cash, check, credit`

要素	例
パイプまたは縦線は複数のオプションのうち 1 つだけを選択できることを意味する。	`cash	check	credit`
省略記号 (...) は、直前の要素を必要な回数だけ繰り返し指定できることを意味する。	`buy thing = price [cash	check	credit] [, thing = price [cash	check	credit]]...` この例では、製品 (thing) を少なくとも 1 つ購入 (buy) し、価格 (price) を指定する必要があります。支払方法を選択できる。角カッコで囲まれた項目の 1 つを選択する。追加品目を、必要な数だけ購入することもできる。各 buy に対して、購入した製品 (thing)、価格 (price)、オプションで支払方法 (cash、check、credit のいずれか) を指定します。

- 次は、オプション句のあるコマンドの構文の例です。

  sp_dropdevice [*device_name*]

  複数のオプションを持つコマンドの例を示します。

  select *column_name*
  from *table_name*
  where *search_conditions*

  構文では、キーワード (コマンド) は通常のフォントで表記し、識別子は小文字で表記します。ユーザが提供するワードは斜体で表記します。

- Transact-SQL コマンドの使用例は次のように表記します。

  ```
  select * from publishers
  ```

- 次は、コンピュータからの出力例です。

```
pub_id    pub_name                city          state
-------   ---------------------   -----------   -----
0736      New Age Books           Boston        MA
0877      Binnet & Hardley        Washington    DC
1389      Algodata Infosystems    Berkeley      CA

(3 rows affected)
```

このマニュアルでは、例に使用する文字はほとんどが小文字ですが、Transact-SQL のキーワードを入力するときは、大文字と小文字は区別されません。たとえば、SELECT、Select、select はすべて同じです。

テーブル名などのデータベース・オブジェクトの大文字と小文字を Adaptive Server が区別するかどうかは、Adaptive Server にインストールされたソート順によって決まります。シングルバイト文字セットを使用している場合は、Adaptive Server のソート順を再設定することによって、大文字と小文字の区別の取り扱い方を変更できます。詳細については、『システム管理ガイド』を参照してください。

アクセシビリティ機能

このマニュアルには、アクセシビリティを重視した HTML 版もあります。この HTML 版マニュアルは、スクリーン・リーダーで読み上げる、または画面を拡大表示するなどの方法により、その内容を理解できるよう配慮されています。

Adaptive Server HTML マニュアルは、連邦リハビリテーション法第 508 条のアクセシビリティ規定に準拠していることがテストにより確認されています。第 508 条に準拠しているマニュアルは通常、World Wide Web Consortium (W3C) の Web サイト用ガイドラインなど、米国以外のアクセシビリティ・ガイドラインにも準拠しています。

注意 アクセシビリティ・ツールを効率的に使用するには、設定が必要な場合もあります。一部のスクリーン・リーダーは、テキストの大文字と小文字を区別して発音します。たとえば、すべて大文字のテキスト (ALL UPPERCASE TEXT など) はイニシャルで発音し、大文字と小文字の混在したテキスト (Mixed Case Text など) は単語として発音します。構文規則を発音するようにツールを設定すると便利かもしれません。詳細については、ツールのマニュアルを参照してください。

Sybase Accessibility (http://www.sybase.com/accessibility) を参照してください。Sybase Accessibility サイトには、第 508 条と W3C 標準に関する情報へのリンクもあります。

不明な点があるときは

Sybase ソフトウェアがインストールされているサイトには、Sybase 製品の保守契約を結んでいるサポート・センタとの連絡担当の方 (コンタクト・パーソン) を決めてあります。マニュアルだけでは解決できない問題があった場合には、担当の方を通して Sybase のサポート・センタまでご連絡ください。

パート 1

システム管理の基本

この章では、Adaptive Server のシステム管理の概要について説明します。

- 「第 1 章 システム管理の概要」では、Sybase システムの構造について説明します。
- 「第 2 章 システム・データベースとオプションのデータベース」では、Adaptive Server システム・データベースの内容と機能について説明します。
- 「第 3 章 システム管理の基礎」では、新たにシステム管理者となった方が実行する必要のある重要な作業について説明します。
- 「第 4 章 Adaptive Server Plug-in for Sybase Central の概要」では、Adaptive Server を管理するためのグラフィカル・ユーザ・インタフェース、Sybase Central を起動および使用する方法について説明します。
- 「第 5 章 設定パラメータ」では、設定パラメータについて説明します。これらのパラメータを sp_configure で設定することによって、Adaptive Server のさまざまな機能を制御できます。
- 「第 6 章 ディスク・リソースについての概要」では、ディスク上のデータベース、テーブル、インデックスの物理的な場所に関連する問題について説明します。
- 「第 7 章 データベース・デバイスの初期化」では、データベース・デバイスの初期化方法およびデバイスをデフォルト・デバイス・プールに割り当てる方法について説明します。
- 「第 8 章 データベース・オプションの設定」では、データベース・オプションの設定方法について説明します。
- 「第 9 章 文字セット、ソート順、言語の設定」では、言語モジュールに含まれるファイルなどの国際化についての問題、および Adaptive Server の言語、ソート順、文字セットの設定方法について説明します。

- 「第 10 章 クライアント／サーバの文字セット変換の設定」では、異機種間環境での Adaptive Server とクライアント間の文字セット変換について説明します。
- 「第 11 章 システムの問題の診断」では、Adaptive Server と Backup Server™ のエラー処理、およびサーバの停止方法とユーザ・プロセスの強制終了方法について説明します。

第 1 章

システム管理の概要

この章では、Adaptive Server のシステム管理についての基本的なトピックを説明します。

Adaptive Server の管理作業

Adaptive Server の管理作業には、次のものがあります。

- Adaptive Server および Backup Server のインストール
- Adaptive Server ログイン・アカウントの作成と管理
- Adaptive Server ユーザに対する役割とパーミッションの付与
- 接続、メモリ、ディスク領域の使用の管理とモニタ
- データベースのバックアップとリストア
- システム上の問題の診断
- パフォーマンスを最大にするための Adaptive Server の設定

さらにシステム管理者は、整合性基準の適用など、データベース設計作業を支援する場合があります。このような役割は、アプリケーション設計者とも共通しています。

システム管理者は一般的に、Adaptive Server 上で実行されるアプリケーションとは直接関係のない作業を主に行いますが、すべてのアプリケーションを最もよく把握できる立場にあります。このためシステム管理者は、アプリケーション設計者に対して Adaptive Server 上の既存のデータに関するアドバイスや、複数のアプリケーションに関係するデータ定義の標準化についての助言などができます。

しかし、アプリケーション独自の機能とそうでない機能の判別が難しい場合もあります。ユーザ・データベースの所有者はこのマニュアルの該当する箇所を参照してください。同様に、システム管理者とデータベース所有者は、『Transact-SQL ユーザーズ・ガイド』(特に、データ定義、ストアド・プロシージャ、トリガに関する章) を参照してください。システム管理者とアプリケーション設計者は、『パフォーマンス&チューニング・シリーズ』を参照してください。

システム管理作業に必要な役割

このマニュアルで説明されている多くのコマンドやプロシージャでは、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を必要とします。コマンドやプロシージャ以外の章は、データベース所有者に関連するものです。

さまざまなセキュリティ関連、管理、運用の作業は、ユーザの役割ごとに次のように分けられます。

- 「システム管理者」(sa) は、デフォルトでは次の役割を持ちます。
  - sa_role
  - sso_role
  - oper_role
  - sybase_ts_role

  システム管理者の作業には次のものがあります。
  - ディスク記憶領域の管理
  - Adaptive Server の自動リカバリ・プロシージャのモニタ
  - 設定可能なシステム・パラメータの変更による Adaptive Server のチューニング
  - システムの問題の診断と報告
  - データベースのバックアップとロード
  - サーバ・ログイン・アカウントの変更と削除
  - システム管理者の役割の付与と取り消し
  - Adaptive Server ユーザへのパーミッションの付与
  - ユーザ・データベースの作成とそのデータベースの所有権の付与
  - パーミッションの付与と取り消しで使用できるグループの設定

- 「システム・セキュリティ担当者」は、次のようなセキュリティ関連作業を実行します。
  - サーバ・ログイン・アカウントの作成 (初期パスワードの割り当てを含む)
  - アカウントのパスワードの変更
  - システム・セキュリティ担当者とオペレータの役割の付与と取り消し
  - ユーザ定義の役割の作成、付与、取り消し
  - サーバ内で別のユーザになり代わる権限の付与
  - パスワードの有効期間の設定
  - ネットワーク・ベースのセキュリティ・サービスを使用するための Adaptive Server の設定
  - 監査システムの管理
- 「オペレータ」は、サーバ全体にわたってデータベースのバックアップとロードを実行します。オペレータの役割によって、1 人のユーザが dump database、dump transaction、load database、load transaction コマンドを使い、各データベースの所有者にならなくても、サーバ上のすべての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。1 つのデータベース内では、データベース所有者またはシステム管理者がこれらの操作を実行できますが、オペレータはすべてのデータベースに対してこれらの操作を実行できます。

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。これらの作業は監査することができ、その責任は役割を付与されているユーザにあります。システム管理者は、任意アクセス制御 (DAC) 保護システムの外部で操作を行います。つまり、システム管理者がオブジェクトにアクセスするときは、Adaptive Server は DAC パーミッションをチェックしません。

さらにオブジェクト所有者には、所有するオブジェクトによって特別なステータスを持つ、2 つのタイプがあります。次のタイプの所有者です。

- データベース所有者
- データベース・オブジェクトの所有者

データベース所有者

「データベース所有者」は、データベースを作成したユーザ、またはデータベースの所有権を譲渡されたユーザです。システム管理者は grant コマンドを使って、ユーザにデータベースを作成する権限を付与します。

データベース所有者は、自分に割り当てられたログイン名とパスワードを使って Adaptive Server にログインし、“dbo” アカウントを所有します。自分が作成したものでないデータベースにログインする場合は、ユーザは通常の自分のユーザ名で識別されます。

データベース所有者は次のことができます。

- システム・プロシージャ sp_adduser を実行して、他の Adaptive Server ユーザがデータベースにアクセスできるようにする。
- grant コマンドを使って、データベース内での、オブジェクト作成やコマンド実行のためのパーミッションを他のユーザに付与する。

データベースにユーザを追加する方法については、「第 14 章 Adaptive Server のログイン、データベース・ユーザ、クライアント接続の管理」を参照してください。ユーザにパーミッションを付与する方法については、「第 17 章 ユーザ・パーミッションの管理」を参照してください。

データベース所有者は、他のユーザが所有しているオブジェクトのパーミッションを自動的には受け取りません。ただしデータベース所有者は、setuser コマンドを使っていつでもデータベース内の別のユーザになり、一時的にそのユーザのパーミッションを使うことができます。setuser と grant コマンドを組み合わせて使うことによって、データベース所有者はデータベース内のどのオブジェクトのパーミッションでも取得することができます。

注意 データベース所有者の役割は非常に強力であるため、システム管理者は、サーバ内のデータベースの所有者をどのユーザにするかを慎重に検討してください。また、システム・セキュリティ担当者は、すべてのデータベース所有者のデータベース・アクティビティを監査するようにしてください。

データベース・オブジェクトの所有者

データベース・オブジェクト所有者とは、データベース・オブジェクトを作成するユーザです。データベース・オブジェクトとは、テーブル、インデックス、ビュー、デフォルト、トリガ、ルール、制約、プロシージャです。ユーザがデータベース・オブジェクトを作成するには、データベース所有者がそのユーザに対して、特定タイプのオブジェクトを作成するためのパーミッションを付与する必要があります。データベース・オブジェクト所有者としての特別なログイン名やパスワードはありません。

データベース・オブジェクト所有者は、create 文を使ってオブジェクトを作成してから、他のユーザにパーミッションを付与します。

データベース・オブジェクト所有者には、そのオブジェクトに対するすべてのパーミッションが自動的に付与されます。システム管理者にも、そのオブジェクトに対するすべてのパーミッションが与えられます。オブジェクトの所有者は、他のユーザがそのオブジェクトにアクセスできるようにするために明示的にパーミッションを付与する必要があります。オブジェクトの所有者が適切なパーミッションを付与しないと、データベース所有者であってもそのオブジェクトを直接使用することはできません。ただし、データベース所有者はいつでも setuser コマンドを使ってオブジェクト所有者をはじめとするデータベース内の別のユーザになり代わることができます。

注意 データベース・オブジェクトがデータベース所有者以外のユーザによって所有されている場合、そのオブジェクトにアクセスするには、システム管理者であってもオブジェクト所有者の名前でオブジェクト名を修飾する (*ownername.objectname*) 必要があります。多数のユーザが同じオブジェクトまたはプロシージャにアクセスする必要がある場合、特にアドホック・クエリでアクセスする場合は、これらのオブジェクトの所有者を “dbo” にしておくと簡単にアクセスできます。

システム管理作業のための *isql* の使用

このマニュアルで説明するシステム管理作業は、コマンド・ライン・ユーティリティ isql を使用して行うことを想定しています。この項では、isql の使用に関する基本的な事項を説明します。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。

このマニュアルで説明している作業の多くは、Sybase Central™ というグラフィック・ツールを使用して実行できます。詳細については、「システム管理作業での Sybase Central の使用」(8 ページ) を参照してください。

isql の起動

ほとんどのプラットフォームで isql を起動するには、オペレーティング・システムのプロンプトで次のコマンドを入力します。*username* は、システム管理者のユーザ名です。

```
isql -Uusername
```

パスワードの入力を要求するプロンプトが表示されます。

注意 パスワードの指定には、isql の -P オプションを使用しないでください。このオプションを使用すると他のユーザにパスワードがわかってしまいます。

コマンド・ライン・モードで isql を使用して、このマニュアルにある Transact-SQL 例の多くを入力できます。

文の入力

isql では、文を複数行に分けて入力することができます。新たな行で “go” を入力すると、isql による文の処理が開始します。次に例を示します。

```
1> select *
2> from sysobjects
3> where type = "TR"
4> go
```

このマニュアルの例では、文と文の間の go コマンドは示していません。例に従って入力する場合に、結果の出力を参照するには go コマンドを入力してください。

文の保存と再使用

Transact-SQL 文を使用してユーザ・データベースとデータベース・オブジェクトを作成または変更する場合は、Transact-SQL 文をその都度保存してください。そのためには、ASCII ファイル形式で文を作成またはコピーするのが最も簡単な方法です。そうすれば、データベースまたはデータベース・オブジェクトを後で作り直す場合に、そのファイルを使用して isql に文を入力できます。

ASCII フォーマットのファイルを指定して isql を実行する場合の構文は次のとおりです。*filename* は、Transact-SQL 文が入力されているファイルのフル・パスとファイル名です。

```
isql -Uusername -ifilename
```

UNIX と他のプラットフォームでファイルをリダイレクトするには、小なり記号 (<) を使用します。

ASCII ファイル内の Transact-SQL 文は有効な構文で記述する必要があります。また、go コマンドを使用する必要があります。

ファイルからコマンドを読み込む場合、次の作業を行う必要があります。

- コマンド・ラインで -P*password* オプションを指定する。または
- 入力ファイルの先頭行に指定するユーザのパスワードを追加する。

システム管理作業での Sybase Central の使用

システム管理作業の多くは、Adaptive Server に付属している Sybase Central というグラフィック・ツールを使用して実行できます。

- データベース・デバイスの初期化
- 設定パラメータの設定
- データベースの空きログ領域の容量表示
- データ定義言語 (DDL) の生成

- ログインの作成
- リモート・サーバの追加
- データベースの作成
- ストアド・プロシージャの作成
- 役割の定義
- データ・キャッシュの追加
- データベース・オプションの設定
- データベースのバックアップとリストア

Sybase Central の Monitor Viewer 機能を使用して、Adaptive Server Monitor™ にアクセスすることもできます。Sybase Central には、詳細なオンライン・ヘルプが用意されています。

Sybase Central の DDL 生成機能を使用して、作業を Transact-SQL スクリプトに記録できます。DDL 生成機能を利用すると、サーバ全体または特定のデータベース内で行う動作をスクリプトに保存できます。

システム・テーブル

master データベースには、Adaptive Server の情報を記録する「システム・テーブル」があります。また、各データベース (master データベースも含む) には、そのデータベース特有の情報を記録するシステム・テーブルがあります。

master データベース (Adaptive Server の制御データベース) 内の Adaptive Server によって作成されるすべてのテーブルは、システム・テーブルと見なされます。また、各ユーザ・データベースが作成されるとき、このようなシステム・テーブルのサブセットも作成されます。システム・テーブルは、「データ辞書」またはシステム・カタログと呼ぶこともあります。

master データベースとそのテーブルは、Adaptive Server のインストール時に自動的に作成されます。ユーザ・データベース内のシステム・テーブルは、create database コマンドを発行した時点で作成されます。システム・テーブル名は、すべて “sys” で始まります。ユーザ・データベース内に、システム・テーブルと同じ名前のテーブルを作成することはできません。システム・テーブルとそのカラムの詳細については、『リファレンス・マニュアル：テーブル』を参照してください。

システム・テーブルの問い合わせ

他のテーブルと同様の方法でシステム・テーブルを問い合わせることができます。例として、データベース内のすべてのトリガ名を返す文を次に示します。

```
select name
from sysobjects
where type = "TR"
```

さらに、Adaptive Server に付属している「ストアド・プロシージャ」(システム・プロシージャ) を利用して、システム・テーブルを簡単に問い合わせることができます。

次のリストは、システム・テーブルからの情報を返すシステム・プロシージャです。

• sp_commonkey	• sp_helpremotelogin
• sp_configure	• sp_help_resource_limit
• sp_countmedatada	• sp_helprotect
• sp_dboption	• sp_helpsegment
• sp_estspace	• sp_helpserver
• sp_help	• sp_helpsort
• sp_helpartition	• sp_helptext
• sp_helpcache	• sp_helpthreshold
• sp_helpconfig	• sp_helpuser
• sp_helpconstraint	• sp_lock
• sp_helpdb	• sp_monitor
• sp_helpdevice	• sp_monitorconfig
• sp_helpgroup	• sp_showcontrolinfo
• sp_helpindex	• sp_showexeclass
• sp_helpjava	• sp_showplan
• sp_helpjoins	• sp_spaceused
• sp_helpkey	• sp_who
• sp_helplanguage	• sp_help_resource_limit
• sp_helplog	

システム・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。

システム・テーブル内のキー

システム・テーブルのプライマリ・キー (主キー)、外部キー、共通キーは、master データベースと model データベース内に定義されます。システム・プロシージャ sp_helpkey を実行すると、定義されたキーに関するレポートを出力できます。2 つのシステム・テーブルをジョインするときに使用できる可能性のあるカラムに関するレポートを出力するには、sp_helpjoins を実行します。

『Adaptive Server システム・テーブル・ダイヤグラム』には、システム・テーブルのカラム間の関係が記載されています。

システム・テーブルの更新

Adaptive Server のシステム・テーブルには、データベースを運用する上で重要な情報が格納されています。通常、システム・テーブルのデータを直接変更する必要はありません。

Sybase の製品の保守契約を結んでいるサポート・センタから指示された場合、または『トラブルシューティング＆エラー・メッセージ・ガイド』やこのマニュアルに指示がある場合を除いて、システム・テーブルは更新しないでください。

システム・テーブルを更新する場合は、システム・テーブルの更新を可能にする sp_configure コマンドを発行しなければなりません。このコマンドが有効な間は、適切なパーミッションを持つユーザであれば誰でもシステム・テーブルを変更できます。システム・テーブルを直接変更する場合の条件は、次のとおりです。

- システム・テーブルの変更は、必ずトランザクション内で行ってください。begin transaction コマンドを発行してから、データ変更コマンドを発行します。
- 変更したいローだけがコマンドの影響を受けたことと、そのデータが正確に変更されたことを確認してください。
- コマンドが正しくない場合は、rollback transaction コマンドを発行します。コマンドが正しい場合は、commit transaction コマンドを発行します。

警告！ どのような状況でも、どのユーザも変更してはならないシステム・テーブルがあります。システム・テーブルには、システム・プロセスによって動的に構築されるものや、コード化された情報を含むもの、あるいは問い合わせを実行してもそのデータの一部しか表示されないものがあります。不用意に通常と違う方法でシステム・テーブルを更新すると、Adaptive Server の実行またはデータベース・オブジェクトに対するアクセスが不可能になる場合があります。また、オブジェクトに対するパーミッションの混乱を招いたり、ユーザ・セッションが終了したりする場合もあります。さらに、システム・テーブルの定義はどのような形であっても変更しないでください。たとえば、制約を含むようにシステム・テーブルを変更しないでください。トリガ、デフォルト、ルールはシステム・テーブルでは許可されていません。トリガを作成しようとしたり、ルールやデフォルトをシステム・テーブルにバインドしようとしたりすると、エラー・メッセージが返されます。

システム・プロシージャ

システム・プロシージャの名前は、すべて “sp_” で始まります。システム・プロシージャは、sybsystemprocs データベース内にありますが、その多くはどのデータベース内でも実行できます。実行するには、そのデータベースからストアド・プロシージャを発行するか、プロシージャ名をデータベース名で修飾します。

Sybase が提供するシステム・プロシージャ (sp_who など) は、*installmaster* インストール・スクリプトを使用して作成されています。最後に実行された *installmaster* のバージョンを判断するには、sp_version を使用します。sp_version 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。

sybsystemprocs 以外のデータベースでシステム・プロシージャを実行した場合は、プロシージャの操作の対象はシステム・プロシージャが実行されたデータベース内のシステム・テーブルになります。たとえば、pubs2 のデータベース所有者が pubs2 から sp_adduser を実行するか、または pubs2..sp_adduser コマンドを発行すると、pubs2..sysusers に新しいユーザが追加されます。ただし、このことは、master データベース内のテーブルだけを更新するシステム・プロシージャには適用されません。

システム・プロシージャに対するパーミッションについては、『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。

システム・プロシージャの使用

「パラメータ」は、ストアド・プロシージャやシステム・プロシージャの引数です。システム・プロシージャのパラメータ値に予約語、句読表記、または埋め込みブランクがある場合は、一重引用符か二重引用符で囲んでください。パラメータがオブジェクト名で、そのオブジェクト名がデータベース名または所有者名で修飾されている場合は、その名前全体を一重引用符か二重引用符で囲んでください。

システム・プロシージャは、連鎖トランザクション・モードと非連鎖トランザクション・モードのどちらかを使用して、セッション中に呼び出すことができます。連鎖モードでは、データ検索文またはデータ修正文の前に暗黙的にトランザクションが開始されます。非連鎖モードでは、トランザクションを完了するために commit transaction 文や rollback transaction 文と対になる明示的 begin transaction 文が必要です。『Transact-SQL ユーザーズ・ガイド』の「第 22 章 トランザクション：データの一貫性およびリカバリ」を参照してください。

master データベースのシステム・テーブルにあるデータを変更するシステム・プロシージャは、トランザクション内では実行できません。このようにすると、データベースのリカバリで問題が発生する可能性があるためです。また、テンポラリ・ワークテーブルを作成するシステム・プロシージャも、トランザクション内では実行できません。

システム・プロシージャの実行時にアクティブなトランザクションがない場合、Adaptive Server は連鎖モードをオフにして、そのプロシージャの実行中は transaction isolation level 1 を設定します。復帰する前に、セッションの連鎖モードと独立性レベル (隔離性水準ともいいます) は元の設定にリセットされます。『Transact-SQL ユーザーズ・ガイド』の「第 22 章 トランザクション：データの一貫性およびリカバリ」を参照してください。

すべてのシステム・プロシージャは、リターン・ステータスをレポートします。たとえば、次の例は、プロシージャが正しく実行されたことを表します。

```
return status = 0
```

システム・プロシージャが正常に実行されない場合、リターン・ステータスは 0 以外の数字になります。

システム・プロシージャ・テーブル

システム・プロシージャは、master データベースと sybsystemdb データベース内の「システム・プロシージャ・テーブル」を使用して、内部システム値 (たとえば、ステータス・ビット) を人間が判読できるフォーマットに変換します。このようなシステム・プロシージャ・テーブルの 1 つである spt_values は、次のようなさまざまなシステム・プロシージャによって使用されます。

• sp_configure	• sp_helpdevice
• sp_dboption	• sp_helpindex
• sp_depends	• sp_helpkey
• sp_help	• sp_helprotect
• sp_helpdb	• sp_lock

spt_values テーブルが更新されるのは、システムがアップグレードされるときだけです。それ以外では更新されません。spt_values テーブルの使用法を確認するには、sp_helptext を実行して、それを参照するシステム・プロシージャのいずれかのテキストを参照してください。

他のシステム・プロシージャ・テーブルには、spt_monitor と spt_committab、およびカタログ・ストアド・プロシージャで必要とするテーブルがあります (spt_committab テーブルは、sybsystemdb データベースにあります)。

また、テンポラリ・テーブルを作成して削除するシステム・プロシージャもあります。たとえば、sp_helpdb は #spdbdesc を、sp_helpdevice は #spdevtab を、sp_helpindex は #spindtab を作成します。

システム・プロシージャの作成

システム・プロシージャの多くは、このマニュアルのシステム・プロシージャに関する章で説明しています。詳細については、『リファレンス・マニュアル：プロシージャ』を参照してください。

システム管理者は、任意のデータベースから実行できるシステム・プロシージャを記述することができます。sybsystemprocs 内にストアド・プロシージャを作成し、“sp_” で始まる名前を付けてください。ストアド・プロシージャの uid は 1 (データベース所有者の uid) にしてください。

システム管理者が作成するシステム・プロシージャのほとんどは、システム・テーブルを問い合わせるものです。システム・テーブルを変更するストアド・プロシージャを作成することはおすすめしません。

システム・テーブルを変更するストアド・プロシージャを作成するには、まず、システム・セキュリティ担当者が allow updates to system tables 設定パラメータを“on”にする必要があります。このパラメータが“on” に設定されている間に作成されたストアド・プロシージャは、allow updates to system tables が “off” に設定されても、常にシステム・テーブルを更新できます。システム・テーブルを更新するストアド・プロシージャの作成方法を次に示します。

1 sp_configure を使用して allow updates to system tables を “on” に設定します。

2 create procedure コマンドを使用してストアド・プロシージャを作成します。

3 sp_configure を使用して allow updates to system tables を “off” に設定します。

警告！ システム・テーブルを変更する場合は特に注意してください。システム・テーブルを変更するプロシージャは、運用データベースではなく、開発データベースやテスト・データベースでテストしてください。

システム拡張ストアド・プロシージャ

拡張ストアド・プロシージャ (ESP) を利用すると、Adaptive Server から外部言語機能を呼び出すことができます。Adaptive Server には定義済みの ESP セットが付属していますが、ユーザが独自の ESP を作成することもできます。システム拡張ストアド・プロシージャの名前はすべて “xp_” で始まります。これらは、sybsystemprocs データベースにあります。

非常に便利なシステム ESP の 1 つに xp_cmdshell があります。これは、Adaptive Server を実行しているシステム上でオペレーティング・システム・コマンドを実行するものです。

システム ESP はシステム・プロシージャとまったく同じように呼び出すことができます。異なる点は、システム ESP は Transact-SQL 文ではなく、手続き型言語コードを実行することです。すべての ESP は、Adaptive Server と同じマシン上で実行される Open Server™ アプリケーションである XP Server™ によって実装されます。XP Server は最初の ESP 実行時に自動的に起動します。

Adaptive Server に付属するシステム ESP 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。

システム ESP の作成

create procedure を使用して、sybsystemprocs データベースにシステム ESP を作成します。システム・プロシージャは自動的に sybsystemprocs データベースに組み込まれます。ESP とその手続き型言語関数には、“xp_” で始まる名前を付けてください。ストアド・プロシージャの uid は 1 (データベース所有者の uid) にしてください。

ESP 作成の一般的な情報については、『Transact-SQL ユーザーズ・ガイド』の「第 18 章 拡張ストアド・プロシージャの使用」を参照してください。

エラー・メッセージのログ

Adaptive Server は、起動されるたびにローカル・エラー・ログ・ファイルに起動情報を書き込みます。新しい Adaptive Server を設定すると、インストール・プログラムが自動的にエラー・ログのロケーションを設定します。エラー・ログのデフォルトのロケーションとファイル名については、使用するプラットフォームの『設定ガイド』を参照してください。

Adaptive Server からのエラー・メッセージの多くはユーザの端末にだけ表示されます。ただし、致命的なエラー・メッセージ (重大度レベル 19 以上)、カーネル・エラー・メッセージ、Adaptive Server からの情報メッセージはエラー・ログ・ファイルに記録されます。

Adaptive Server は、サーバ・プロセスが停止されるまではエラー・ログ・ファイルをオープンした状態に保ちます。古いメッセージを削除してエラー・ログのサイズを減らすには、その前に Adaptive Server プロセスを停止してください。

注意　Windows など一部のプラットフォームでは、Adaptive Server はオペレーティング・システムのイベント・ログにもエラー・メッセージを記録します。詳細については、使用しているプラットフォームの『インストール・ガイド』と『設定ガイド』を参照してください。

Adaptive Server との接続

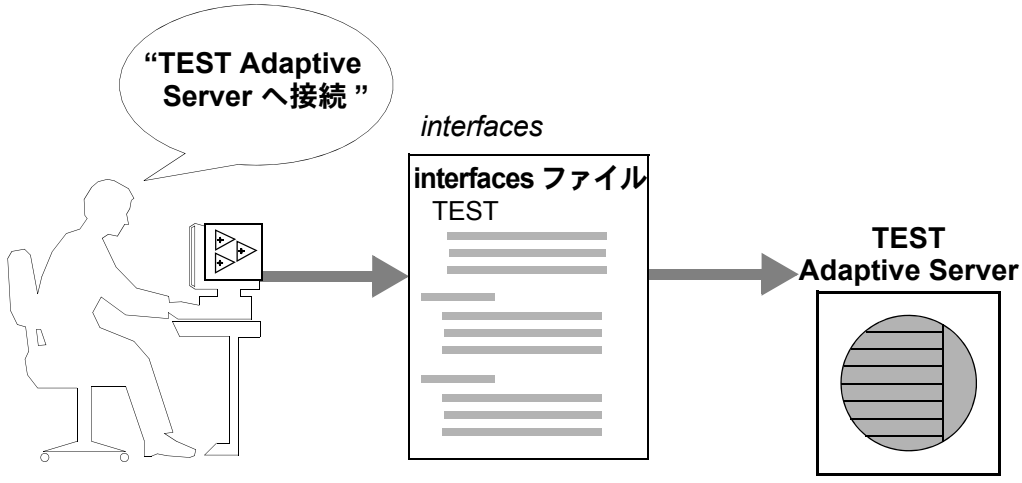
Adaptive Server は、別の Adaptive Server、Open Server アプリケーション、ネットワーク上のクライアント・ソフトウェアと通信できます。クライアントは 1 つ以上のサーバと通信でき、サーバはリモート・プロシージャ・コールを使用して別のサーバと通信できます。これらが対話するには、それぞれがネットワーク上での相手のロケーションを知る必要があります。このネットワーク・サービス情報は *interfaces* ファイルに保管されます。

interfaces ファイル

この *interfaces* ファイルの名前は、オペレーティング・システムによって、*interfaces*、*interface*、または *sql.ini* となります。

interfaces ファイルには認識されているすべてのサーバ名とアドレスがリストされています。クライアント・プログラムを使用してサーバと接続するとき、図 1-1 に示すように、プログラムは *interfaces* ファイル内でサーバ名を探し、そのアドレスを使用してサーバに接続します。

図 1-1: *Adaptive Server* との接続



interfaces ファイルの名前、ロケーション、内容はオペレーティング・システムによって異なります。また、*interfaces* ファイル内の Adaptive Server アドレスのフォーマットもネットワーク・プロトコルによって異なります。

Adaptive Server のインストール時に、1 つ以上のネットワーク・プロトコルによる Adaptive Server へのローカル接続用に使用できる簡単な *interfaces* ファイルが作成されます。ユーザがネットワークを通して Adaptive Server に接続できるように *interfaces* ファイルを変更してユーザに配布するのは、システム管理者です。*interfaces* ファイルの詳細については、使用しているプラットフォームの『設定ガイド』を参照してください。

interfaces ファイルとネットワーク・リスナの詳細については、『パフォーマンス&チューニング・シリーズ：基本』の「第 2 章 ネットワークとパフォーマンス」を参照してください。

ディレクトリ・サービス

ディレクトリ・サービスは、ネットワーク・サービス情報の作成、修正、取得を管理します。ディレクトリ・サービスは、プラットフォームやサードパーティのベンダによって提供されるものであり、Adaptive Server とは別に購入してインストールする必要があります。ディレクトリ・サービスの例には、レジストリと分散コンピューティング環境 (DCE) があります。

$SYBASE/$SYBASE_OCS/config/libtcl.cfg ファイルは、Sybase が提供する設定ファイルで、サーバとクライアントが次の項目を決定するときに使用します。

- 使用するディレクトリ・サービス
- そのディレクトリ・サービスのドライバの場所

ディレクトリ・サービスがまったくインストールされていない場合や、*libtcl.cfg* ファイルにエントリがまったくない場合は、Adaptive Server は *interfaces* ファイルをデフォルトとして使用して、ネットワーク・サービス情報を取得します。

システム管理者は、操作環境に応じて *libtcl.cfg* ファイルを修正する必要があります。

ディレクトリ・サービスには、プラットフォーム固有のものも、いくつかの異なるプラットフォーム上で使用できるものもあります。プラットフォーム固有のディレクトリ・サービスの設定の詳細については、使用するプラットフォームの『設定ガイド』を参照してください。

ディレクトリ・サービスとしての LDAP

LDAP (Lightweight Directory Access Protocol) は、ディレクトリ・サービスへの業界標準のアクセス方法です。ディレクトリ・サービスを使用すると、コンポーネントは LDAP サーバから情報を DN (識別名) で検索できます。LDAP サーバは、企業またはネットワーク上で使用されるサーバ、ユーザ、ソフトウェアの情報を格納したり管理したりします。

LDAP サーバは、Adaptive Server やクライアントを実行しているプラットフォームとは別のプラットフォームに配置できます。LDAP は、クライアントとサーバが交換するメッセージの通信プロトコルと内容を定義します。メッセージとは、読み取り、書き込み、クエリのクライアント要求やサーバの応答など、メタデータ (データに関するデータ) を含むオペレータです。

LDAP サーバに格納され、取得が可能な情報は、次のとおりです。

- Adaptive Server に関する情報 (IP アドレス、ポート番号、ネットワーク・プロトコルなど)

- セキュリティ・メカニズムとフィルタ
- 高可用性コンパニオン・サーバ名
- Adaptive Server にユーザがアクセスするための認証情報

  Adaptive Server にログインするユーザを認証するには、*syslogins* ディレクトリに格納されている情報を使用することも、単一のログインとパスワードを企業全体で使用可能にする集中型の LDAP サーバを使用することもできます。「第 14 章 Adaptive Server のログイン、データベース・ユーザ、クライアント接続の管理」を参照してください。

LDAP サーバの設定時に、次のアクセス制限を指定できます。

- 匿名認証 － すべてのユーザがあらゆる情報にアクセスできます。
- ユーザ名とパスワードによる認証 － Adaptive Server は、次のファイルで指定されているデフォルトのユーザ名とパスワードを使用します。
  - UNIX、32 ビット － *$SYBASE/$SYBASE_OCS/config/libtcl.cfg*
  - UNIX、64 ビット － *$SYBASE/$SYBASE_OCS/config/libtcl64.cfg*
  - Windows － *%SYBASE%¥%SYBASE_OCS%¥ini¥libtcl.cfg*

ユーザ名とパスワードによる認証のプロパティによって、LDAP サーバとのセッション接続が確立され、終了します。

注意 *libtcl.cfg* に格納されている、認証目的で LDAP サーバに渡されるデフォルトのユーザ名とパスワードは、Adaptive Server へのアクセスに使用するユーザ名とパスワードとはまったく別のものです。このデフォルトのユーザ名とパスワードは、管理作業を実行するために LDAP サーバにアクセスするためのものです。

LDAP サーバを *libtcl.cfg* ファイルまたは *libtcl64.cfg* ファイル (*libtcl*.cfg* ファイルと総称) で指定する場合は、サーバ情報には LDAP サーバからのみアクセスできます。Adaptive Server は *interfaces* ファイルを無視します。

複数のディレクトリ・サービスが 1 つのサーバでサポートされる場合は、その検索の順序は *libtcl*.cfg* に指定されます。検索順は dataserver コマンド・ライン・オプションでは指定できません。

複数のディレクトリ・サービス

LDAP サービスは、どのようなタイプでも (実際のサーバであっても、その他の LDAP サービスへのゲートウェイであっても)、LDAP サーバと呼ばれます。

高可用性を確保するフェールオーバ保護のために、*libtcl*.cfg* ファイルに複数のディレクトリ・サービスを指定できます。リストにあるディレクトリ・サービスのすべてが LDAP サーバである必要はありません。

次の例では、*test:389* への接続が失敗した場合には、指定されたディレクトリ情報ツリー (DIT) ベースを持つ DCE ドライバへのフェールオーバが発生します。この接続も失敗すると、*huey:11389* 上の LDAP サーバに接続しようとします。DIT ベースのフォーマットはベンダによって異なります。

```
[DIRECTORY]
   ldap=libdldap.so ldap://test:389/dc=sybase,dc=com
   dce=libddce.so ditbase=/.:/subsys/sybase/dataservers
   ldap=libdldap.so ldap://huey:11389/dc=sybase,dc=com
```

詳細については、『Open Client Client-Library/C プログラマーズ・ガイド』と『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。

LDAP ディレクトリ・サービスと Sybase *interfaces* ファイルの違い

LDAP サーバで使用するために、LDAP ドライバでディレクトリ・サービスを実装します。LDAP インフラストラクチャの構成は、次のとおりです。

- 従来の Sybase *interfaces* ファイルに代わる、ネットワーク・ベースのしくみ
- ユーザ、ソフトウェア、リソース、ネットワーク、ファイルなどの情報を階層構造で表した単一のビュー

表 1-1 は、Sybase *interfaces* ファイルと LDAP サーバの違いをまとめたものです。

表 *1-1: interfaces* ファイルと *LDAP* ディレクトリ・サービスの違い

interfaces ファイル	ディレクトリ・サービス
プラットフォーム固有	プラットフォームに依存しない
Sybase インストール環境ごとに異なった構造	統一された階層構造
マスタ・エントリとクエリ・エントリが別々に存在する	各サーバの 1 つのエントリにクライアントとサーバの両方がアクセスできる
サーバのメタデータを保存できない	サーバのメタデータを保存できる

パフォーマンス

LDAP サーバを使用する場合は、*interfaces* ファイルを使用した場合よりもパフォーマンスが低下することがあります。これは、LDAP サーバの場合、ネットワーク接続を確立してデータを取得する必要があり、そのために時間を要するからです。この接続は Adaptive Server を起動したときに行われるので、パフォーマンスに違いがある場合はログイン時にわかります。通常のシステム負荷では、パフォーマンスの低下を感じることはありません。特に短い間隔で接続を繰り返す場合など、接続数の増加によってシステム負荷が高まると、LDAP サーバを使用した場合と従来の *interfaces* ファイルを使用した場合とで全体的なパフォーマンスにはっきりとした違いが現れることがあります。

Adaptive Server で使用できるセキュリティ機能

表 1-2 は、Adaptive Server で使用できる主なセキュリティ機能の概要を示しています。Adaptive Server のセキュリティを設定する方法については、このマニュアルの第 2 巻を参照してください。

表 *1-2:* 主要なセキュリティ機能

セキュリティ機能	説明	参照
識別と認証の制御	承認されたユーザだけがシステムにログインできるようにする。Adaptive Server は、パスワードベースのログイン認証の他に、Kerberos、LDAP、PAM (Pluggable Authentication Modules) による外部認証もサポートしている。	「識別と認証」(373 ページ)
任意アクセス制御 (DAC)	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機能。通常は grant コマンドと revoke コマンドを使用する。この種の制御は、オブジェクトの所有者が自由に設定できる。	「任意アクセス制御」(374 ページ)
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する。Adaptive Server には、システム管理者やシステム・セキュリティ担当者などの「システム標準の役割」と呼ばれる、事前に定義された役割がある。また、システム・セキュリティ担当者が「ユーザ定義の役割」と呼ばれる追加の役割を定義できる。	「役割の分担」(375 ページ)
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する機能。1 つのオプションを設定するだけで、サーバ全体にわたる一連のセキュリティ関連イベントを監査することもできる。	「責任範囲の明確化のための監査」(376 ページ)
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 (Secure Sockets Layer) による暗号化を使用して、データの機密性を保持する。アクティブでないデータは、パスワードで保護されたデータベース・バックアップによって機密性を保持される。	「データの機密保持」(377 ページ)

第 2 章

システム・データベースとオプションのデータベース

この章では、すべての Adaptive Server システムに存在するシステム・データベースについて説明します。また、ユーザがインストール可能な Sybase が提供するオプションのデータベースや、Sybase 製品の保守契約を結んでいるサポート・センタが診断の目的でインストールする sybdiag データベースについても説明します。

システム・データベースの概要

Adaptive Server をインストールすると、デフォルトで次のシステム・データベースもインストールされます。

- master データベース
- model データベース
- システム・プロシージャ・データベース sybsystemprocs
- 2 フェーズ・コミット・トランザクション・データベース sybsystemdb
- テンポラリ・データベース tempdb

オプションで次のデータベースもインストールできます。

- 監査データベース sybsecurity
- サンプル・データベース pubs2 と pubs3

- dbcc データベース dbccdb
- Job Scheduler データベース sybmgmtdb

master、model、sybsystemprocs、tempdb、sybmgmtdb の各データベースのインストールの詳細については、使用するプラットフォームの『インストール・ガイド』を参照してください。dbccdb のインストールについては、『システム管理ガイド 第 2 巻』の「10 章 データベースの一貫性の検査」を参照してください。Job Scheduler の使用方法については、『Job Scheduler ユーザーズ・ガイド』を参照してください。

master、model、sybsystemdb、テンポラリの各データベースは、インストール中に指定したマスタ・デバイス上にすべて常駐します。master データベースは全部がマスタ・デバイスに入っていて、他のデバイスに拡張することはできません。その他のデータベースやユーザ・オブジェクトは、すべて他のデバイス上に作成してください。

警告！ ユーザ・データベースをマスタ・デバイスに保管しないでください。保管してしまうと、システム・データベースとマスタ・デバイスに保管されたユーザ・データベースのリカバリが困難になります。

---

sybsecurity データベースと sybmgmtdb データベースは、専用のデバイスとセグメントにインストールします。使用しているプラットフォームの『インストール・ガイド』を参照してください。

sybsystemprocs データベースは、ユーザが選択したデバイスにインストールできます。pubs2 用と pubs3 用のインストール・スクリプトを変更して、sybsystemprocs 用に作成したデバイスを共有できます。

installjsdb スクリプト (*$SYBASE/ASE-15_0/scripts* にあります) を使用して sybmgmtdb データベースをインストールします。*installjsdb* は *sybmgmtdev* という名前のデバイスを探し、このデバイスに sybmgmtdb データベースとそのテーブル、ストアド・プロシージャを作成します。sybmgmtdb データベースが既に存在する場合、*installjsdb* は既存のデータベースに Job Scheduler テーブルとストアド・プロシージャを作成します。*sybmgmtdev* デバイスも sybmgmtdb データベースも見つからない場合、*installjsdb* はマスタ・デバイスに sybmgmtdb データベースを作成します。ただし、マスタ・デバイスから sybmgmtdb データベースを削除することを強くおすすめします。

installpubs2 スクリプトと *installpubs3* スクリプトは、create database 文内ではデバイスを指定しないため、デフォルト・デバイスに作成されます。インストール時には、マスタ・デバイスがデフォルト・デバイスになります。デバイスを変更するには、スクリプトを編集するか、「第 7 章 データベース・デバイスの初期化」の指示に従ってください。

master データベース

master データベースは、Adaptive Server のオペレーションを制御し、ユーザ・データベースとそれに関連するデータベース・デバイスについての情報をすべて保管します。表 2-1 は、master データベースに記録される情報を示します。

表 *2-1: master* データベースに保管される情報

情報	システム・テーブル
ユーザ・アカウント	syslogins
リモート・ユーザ・アカウント	sysremotelogins
このサーバが対話できるリモート・サーバ	sysservers
進行中のプロセス	sysprocesses
設定可能な環境変数	sysconfigures
システム・エラー・メッセージ	sysmessages
Adaptive Server 上のデータベース	sysdatabases
各データベースに割り付けられている記憶領域	sysusages
システムにマウントされたテープとディスク	sysdevices
アクティブ状態のロック	syslocks
文字セット	syscharsets
言語	syslanguages
サーバ全体に適用される役割を持つユーザ	sysloginroles
サーバの役割	syssrvroles
オンラインの Adaptive Server エンジン	sysengines

master データベースは、ユーザ・データベースとデバイスに関する情報を保管するので、create database、alter database、disk init、disk refit、disk reinit、ディスク・ミラーリングの各コマンドは master データベース内から発行する必要があります。

master データベースの最小サイズは、サーバの論理ページ・サイズによって異なります。master データベースは 6656 以上の論理ページを持つため、論理ページごとの最小物理サイズは次のようになります。

- 2K ページ － 13MB
- 4K ページ － 26MB
- 8K ページ － 52MB
- 16K ページ － 104MB

master でのオブジェクト作成の制御

Adaptive Server をインストールした直後は、master データベースにオブジェクトを作成できるのはシステム管理者だけです。システム管理者は、暗黙のうちに、使用するデータベースの所有者 “dbo” になるからです。master データベースに作成するオブジェクトは、システム管理のためだけに使用してください。一般ユーザが master にオブジェクトを作成できないように、パーミッションを設定してください。

警告！ master 内にはユーザ・オブジェクトを置かないでください。master 内にユーザ・オブジェクトを置くと、トランザクション・ログがすぐにいっぱいになってしまいます。トランザクション・ログが領域を完全に使い果たしてしまうと、dump transaction コマンドを使用して master 内の領域を解放できなくなります。

---

sp_modifylogin を使用してユーザのデフォルト・データベース (ユーザがログイン時に接続するデータベース) を変更する方法もあります。「データベースへのユーザの追加」(384 ページ) を参照してください。

システム・プロシージャは master データベースではなく、sybsystemprocs データベースに作成します。

master のバックアップとシステム・テーブルのコピー

Adaptive Server 上でのハードウェアやソフトウェアの障害に備えて、次のタスクを行います。

- master データベースとすべてのユーザ・データベースの頻繁なバックアップ。詳細については、「master の最新のバックアップの保持」(39 ページ) を参照してください。また、master データベースのリカバリ処理の概要については、『システム管理ガイド 第 2 巻』の「第 13 章 システム・データベースのリストア」を参照してください。
- システム・テーブルのコピーを (なるべくオフラインで) 保存します。sysusages、sysdatabases、sysdevices、sysloginroles、syslogins のコピーを確認します。「システム・テーブルのオフライン・コピーの保存」(40 ページ) を参照してください。これらのスクリプトのコピーが保存されていれば、ハード・ディスクの故障などの障害によってデータベースが使用できなくなった場合でも、『システム管理ガイド第 2 巻』の「第 13 章 システム・データベースのリストア」で説明している手順を使用してリカバリできます。スクリプトの最新コピーが保存されていない場合は、master データベースが損傷を受けた場合に Adaptive Server のリカバリが非常に難しくなります。

model データベース

Adaptive Server には、model データベースが含まれています。このデータベースは、新しいユーザ・データベース用のテンプレート (プロトタイプ) として使用されます。create database コマンドが実行されるたびに、Adaptive Server は model データベースのコピーを作成して、新しいデータベースのサイズを create database コマンドで指定されたとおりに拡張します。

注意　新しいデータベースは、少なくとも、model データベースと同等の大きさでなければなりません。

---

model データベースには、それぞれのユーザ・データベースに必要なシステム・テーブルがあります。model を変更することにより、新しく作成されるデータベースの構造をカスタマイズできます。model に対して行った変更は、新しいデータベースにすべて反映されます。システム管理者が通常行う model への変更は次のとおりです。

- ユーザ定義のデータ型、ルール、またはデフォルトの追加。
- Adaptive Server 上のすべてのデータベースにアクセスできるユーザの追加。
- デフォルト権限、特に “guest” アカウントのデフォルト権限の付与。
- select into/bulkcopy/pllsort などのデータベース・オプションの設定。これらの設定は、すべての新しいデータベースに反映されます。model のオプションのデフォルト設定は off です。「第 8 章 データベース・オプションの設定」を参照してください。

通常、model データベースを変更するパーミッションは、ほとんどのユーザには与えられていません。model データベースの内容はすべて Adaptive Server によって新しいユーザ・データベースにコピーされるので、読み込みパーミッションの付与も、あまり意味がありません。

model データベースは、tempdb よりも大きくすることはできません。デフォルトでは、model データベースのサイズは 6 アロケーション・ユニット (1 アロケーション・ユニットは 256 論理ページ) です。tempdb よりも大きくなるように model のサイズを拡張しようとすると、エラー・メッセージが表示されます。

注意　model データベースのバックアップ・コピーを取っておいてください。また、model を変更するたびに dump database をバックアップします。メディア障害が発生した場合は、ユーザ・データベースの場合と同じ方法で model をリストアします。

---

sybsystemprocs データベース

Sybase のシステム・プロシージャは、sybsystemprocs データベースに保管されています。データベースのユーザがシステム・ストアド・プロシージャ (sp_ で始まる名前のプロシージャ) を実行すると、Adaptive Server は最初にユーザの現在のデータベース内からそのプロシージャを探します。現在のデータベース内にその名前のプロシージャが存在しない場合は、sybsystemprocs 内で探します。sybsystemprocs 内にもそのプロシージャがない場合は、master 内でそのプロシージャを探します。

プロシージャによってシステム・テーブルが変更される (たとえば、sp_adduser によって sysusers テーブルが変更される) と、プロシージャを実行したデータベース内でその変更が行われます。

システム・プロシージャのデフォルト・パーミッションを変更するには、sybsystemprocs でのパーミッションを変更します。

注意 sybsystemprocs を変更する場合は、データベースをバックアップしてください。

---

tempdb データベース

Adaptive Server には、tempdb という「テンポラリ・データベース」があります。テンポラリ・データベースは、テンポラリ・テーブルやその他の一時的な作業に使用される記憶領域です。tempdb の領域は、サーバ上の全データベースの全ユーザ間で共有されます。

tempdb のデフォルト・サイズは、サーバの論理ページ・サイズが 2K、4K、8K、16K のいずれであるかによって決まります。一部のアクティビティのために、tempdb のサイズを大きくしなければならなくなる場合があります。

- 大規模なテンポラリ・テーブル。
- テンポラリ・テーブル上での多数のアクティビティ。これによって、tempdb のログがいっぱいになります。
- 大規模な、または同時に行われる多数のソート。サブクエリや group by による集約によっても、tempdb 内にアクティビティが発生します。

tempdb のサイズを拡張するには、alter database コマンドを使います。tempdb は、最初はマスタ・デバイス上に作成されます。マスタ・デバイスからでも、その他のデータベース・デバイスからでも tempdb に領域を追加できます。

update index statistics を大きなテーブルに対して実行するときに、tempdb がコマンドを処理するのに十分な大きさでない場合、コマンドは失敗します (エラー番号 1105)。

システム・テンポラリ・データベース (つまり、tempdb) の他に、複数のテンポラリ・データベースを作成し、管理することができます。複数のテンポラリ・データベースを使用すると、システム・カタログやシステム tempdb のログに対する競合が減少します。

テンポラリ・テーブルの作成

テンポラリ・テーブルを作成するときや、テンポラリ・データベース内の記憶領域が必要となるコマンドを実行するときも、特別なパーミッションは必要ありません。

テンポラリ・テーブルを作成するには、create table 文中でテーブル名の前にシャープ記号 (#) を付けるか、または名前のプレフィクス “tempdb..” を指定します。

シャープ記号を付けて作成されたテンポラリ・テーブルにアクセスできるのは、Adaptive Server の現在のセッションだけです。したがって、他のセッションのユーザはアクセスできません。このような共有できないテンポラリ・テーブルは、各セッションの終了時に破壊されます。テーブル名の最初の 13 バイト (シャープ記号 (#) も含む) は、ユニークでなければなりません。Adaptive Server は、このようなテーブル名に 17 バイトの数字サフィックスを割り当てます (tempdb..sysobjects を問い合わせれば、このサフィックスを参照できます)。

“tempdb..” プレフィクスを付けて作成されたテンポラリ・テーブルは、tempdb に保管され、Adaptive Server セッション間で共有できます。Adaptive Server は、このようにして作成されたテンポラリ・テーブルの名前を変更しません。このテンポラリ・テーブルは、Adaptive Server が再起動されるか、テーブルの所有者が drop table を使用してテーブルを削除するまで存在します。

システム・プロシージャはテンポラリ・テーブル上で機能します。ただし、システム・プロシージャを tempdb から使用した場合にかぎります。

ストアド・プロシージャが作成したテンポラリ・テーブルは、プロシージャが終了すると削除されます。セッション終了前に、明示的にテンポラリ・テーブルを削除することもできます。

警告！ 他のユーザやセッションとの間でテーブルを共有する場合以外は、ストアド・プロシージャ内から “tempdb..” プレフィクスを付けたテンポラリ・テーブルを作成しないでください。

Adaptive Server の再起動のたびに model が tempdb にコピーされ、これによってデータベースがクリアされます。テンポラリ・テーブルはリカバリできません。

sybsecurity データベース

Adaptive Server の監査のシステムを格納している sybsecurity データベースには次のものが含まれます。

- 監査証跡が保存されるシステム・テーブル sysaudits_01、sysaudits_02、... sysaudits_08
- グローバルな監査オプションを記述したローが保存されている sysauditoptions テーブル
- model から抽出された、その他すべてのデフォルト・システム・テーブル

「第 18 章 監査」を参照してください。

sybsystemdb データベース

sybsystemdb データベースは、分散トランザクションに関する情報を格納します。Adaptive Server バージョン 12.0 以降では、リモート・プロシージャ・コール (RPC) またはコンポーネント統合システム (CIS) を使用してリモート・サーバにトランザクションを送信するためのトランザクション・コーディネーション・サービスを実行することができます。分散トランザクションに参加しているリモート・サーバに関する情報は、syscoordinations テーブルに格納されます。

注意 分散トランザクション管理 (DTM) サービスの詳細については、『Adaptive Server 分散トランザクション管理機能の使用』と『インストール・ガイド』を参照してください。

---

sybsystemdb データベースには、Sybase 2 フェーズ・コミット・プロトコルを使用する SYB2PC トランザクションに関する情報も格納されています。それぞれの 2 フェーズ・コミット・トランザクションに関する情報を格納し、その完了ステータスを追跡する spt_committab テーブルは、sybsystemdb データベースに格納されています。

2 フェーズ・コミット・トランザクションの情報と、sybsystemdb データベースの作成方法については、使用するプラットフォームの『設定ガイド』を参照してください。

sybmgmtdb データベース

ジョブ、スケジュール、スケジュール・ジョブ情報、Job Scheduler タスクで内部処理のために必要なデータは、sybmgmtdb データベースに格納されます。また、実行したそれらのタスクの結果と出力も、sybmgmtdb データベースに格納されます。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。

pubs2 と *pubs3* のサンプル・データベース

サンプル・データベース pubs2 と pubs3 のインストールは任意です。これらのデータベースは、Adaptive Server の学習ツールとして用意されています。Adaptive Server のマニュアルに記載されている例のほとんど (pubs3 データベースを使用していることが明記されている例は除く) で、pubs2 サンプル・データベースが使用されています。pubs2 と pubs3 のインストール方法については、使用するプラットフォームの『インストール・ガイド』を参照してください。サンプル・データベースの内容については、『Transact-SQL ユーザーズ・ガイド』を参照してください。

サンプル・データベースの管理

サンプル・データベースには “guest” ユーザ・ログインが登録されているので、認可された Adaptive Server のユーザであれば誰でも “guest” ユーザとしてそのデータベースにアクセスできます。pubs2 と pubs3 では、ユーザ・テーブルを選択 (select)、挿入 (insert)、更新 (update)、削除 (delete) するためのパーミッションなど、幅広い権限が “guest” ログインに与えられています。「第 14 章 Adaptive Server のログイン、データベース・ユーザ、クライアント接続の管理」を参照してください。

pubs2 データベースと pubs3 データベースのサイズは、サーバの論理ページ・サイズが 2 K、4 K、8 K、16K のいずれかであるかによって決まります。可能であれば、新しいユーザには pubs2 と pubs3 の変更を加えていないコピーを提供してください。そうすることにより、新しいユーザが他のユーザの加えた変更に戸惑うことがなくなります。特定のデータベース・デバイス上に pubs2 と pubs3 を置く場合は、インストール・スクリプトを編集してからデータベースをインストールしてください。

空き領域の問題がある場合は、begin transaction コマンドを発行してからサンプル・データベースを更新するように、ユーザに指示を与えてください。こうすると、サンプル・データベースの更新が終わった後で、rollback transaction コマンドを発行して変更を元に戻すことができます。

pubs2 image データ

Adaptive Server には、pubs2 データベースに image データをインストールするためのスクリプトがあります (pubs3 では image データを使用しません)。image データは 6 つのピクチャで構成され、PICT、TIF、Sun raster の各ファイル・フォーマットが 2 つずつあります。Sybase は、image データを表示するためのツールを提供していません。イメージを表示するには、データベースから image データを抽出した後で、適切なスクリーン・グラフィック・ツールを使用してください。

image データを pubs2 にインストールする方法については、使用するプラットフォームの『インストール・ガイド』を参照してください。

dbccdb データベース

dbcc checkstorage を実行すると、「ターゲット・データベース」の設定情報、オペレーション・アクティビティ、そのオペレーションの結果が dbccdb データベースに記録されます。このデータベースには、dbccdb の作成と管理を行ったり、dbcc checkstorage オペレーションの結果についてのレポートを生成したりする dbcc ストアド・プロシージャが格納されます。『システム管理ガイド第 2 巻』の「第 10 章 データベースの一貫性の検査」を参照してください。

sybdiag データベース

Sybase 製品の保守契約を結んでいるサポート・センタは、デバッグのために、ご使用のシステム上に sybdiagdb データベースを作成することがあります。このデータベースには診断設定データが保持されていて、顧客が使用することはできません。

インストール・スクリプトのバージョンの確認

sp_version を使用して、Adaptive Server にインストールされているスクリプト (*installmaster*、*installdbccdb* など) の現在のバージョン、それらのスクリプトが正常に実行されたかどうか、および、実行に要した時間を確認できます。

sp_version の構文は次のとおりです。

```
sp_version [script_file [, "all"]]
```

各パラメータの意味は、次のとおりです。

- *script_file* は、インストール・スクリプトの名前 (デフォルト値は NULL) です。
- all はスクリプトについての詳細な情報 (実行された日付や実行に要した時間など) をレポートします。

sp_version をパラメータなしで実行すると、スクリプトの全情報をレポートします。

次の例では、実行されたインストール・スクリプト、それらのインストール・スクリプトの実行時刻、終了時刻についてレポートします。

```
sp_version null, 'all'
Script           Version
Status
-----------      ------------------------------------------------------------
-----------
installmaster    15.0/EBF XXXXX/B/Sun_svr4/OS 5.8/asemain/1/32-bit/OPT/Thu Sep 23
22:12:12 2004
Complete [Started=Sep 24 2004  3:39PM]-[Completed=Sep 24 2004  3:45PM
```

第 3 章

システム管理の基礎

この章では、次のことについて説明します。

- 新しいシステム管理者に対する重要なトピックの説明
- システム管理者向けの Sybase のマニュアル内の情報の参照先

この章は、経験豊富なシステム管理者にとっても、継続的な管理アクティビティの整理に役立ちます。

論理ページ・サイズ

データベース・オブジェクトは、論理ページを使用して構築されます。データベースとそれに関連する任意のオブジェクトは、同じ論理ページ・サイズを使用します。つまり、複数の論理ページ・サイズを使用するサーバを作成することはできません。Adaptive Server では、マスタ・デバイスと master データベースを作成するときに論理ページ・サイズを 2K、4K、8K、または 16K とすることができます。ただし、1 台のサーバ・インストール環境で使用できるのは、この 4 種類の論理ページ・サイズのうちの 1 つだけです。サーバ内のすべてのデータベース、および、各データベースにあるすべてのオブジェクトに、同じ論理ページ・サイズが使用されます。たとえば、サーバの論理ページ・サイズが 4K の場合は、ページによっては最初の 2K を超える部分を使用しないことがあるとしても、すべてのページが 4K でなければなりません。

dataserver -z を使用してマスタ・デバイスを作成するときに、ページ・サイズを選択します。

dataserver コマンド (マスタ・デバイスの作成に使用されるコマンド) 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。論理ページ・サイズの詳細については、『システム管理ガイド 第 2 巻』の「第 3 章 メモリの設定」を参照してください。

テスト・サーバの使用方法

Sybase では、テスト用または開発用の Adaptive Server をインストールして使用し、サーバ管理の経験を積んでからそのサーバを削除し、実際の運用サーバを作成することをおすすめします。このようにしてテスト・サーバを使用すれば、さまざまな設定のプランとテストが簡単にでき、間違えた場合でもリカバリの苦労がほとんどありません。実際の運用サーバの再起動や運用データベースの作り直しの必要がなければ、新しい機能のインストールと管理の方法も、ずっと学びやすくなります。

テスト・サーバを使用する場合は、Adaptive Server のインストールまたはアップグレードから始まるサーバ設定作業全体でテスト・サーバを使用することをおすすめします。最終的な運用システムに関する非常に重要な決定はこの手順の中で行われます。次の項では、テスト・サーバがシステム管理者にとってどのように役立つかを説明します。

リソースの計画

テスト・サーバを使用することによって、システムに必要とされる最終的なリソース要件の計画を立てることができ、予想していなかったリソースの不足を発見できます。

特にディスク・リソースは運用システムの最終的な設計に劇的な影響を及ぼすことがあります。たとえば、あるデータベースではメディア障害が発生した場合にノンストップ・リカバリが必要であると決定したとします。このような状況では、重要なデータベースをミラーリングするのに追加データベース・デバイスを 1 つ以上設定する必要があります。テスト・サーバでこのようなリソース要件を発見すれば、データベースの利用者に影響を与えることなく、データベースとテーブルの物理的なレイアウトを変更できます。

テスト・サーバを使用すると、異なるハードウェア設定を使用して Adaptive Server とユーザ・アプリケーションのベンチマーク・テストを行うこともできます。この場合は、Adaptive Server レベルとオペレーティング・システム・レベルの両方で物理リソースの最適な設定を決定してから、システム全体をオンラインにして通常使用を開始します。

パフォーマンスの目標の達成

パフォーマンスの目標のほとんどは、データベースの設計と設定を慎重に計画しなければ達成できません。たとえば、特定のテーブルの挿入処理と I/O パフォーマンスがボトルネックであることを発見したとします。この場合、そのテーブルを専用セグメント上に作成し直してテーブルを分割するのが最善の策と考えられます。しかし、そのような変更は運用システムにとっては混乱のもとであり、設定パラメータの変更だけであっても Adaptive Server を再起動することになります。

Sybase 製品のインストール時の考慮事項

Adaptive Server と他の Sybase 製品のインストールを行うときに、システム管理者が担当責任者になることがあります。インストールを担当する場合は、次の指標を使用して処理に役立ててください。

製品の互換性のチェック

新しい製品をインストールするときや既存の製品をアップグレードするときは、その前に、製品に添付されている『リリース・ノート』を読み、システムに影響する互換性の問題について理解してください。互換性の問題は、ハードウェアとソフトウェア間、同じソフトウェアの異なるリリース・レベル間で発生する可能性があります。前もって『リリース・ノート』を読んでおくことによって、互換性に関する既知の問題の解決に費やす時間を節約し、無用な推測を避けることができます。リリース・ノートに記載されている既知の問題には、特に注意してください。

Adaptive Server のインストールまたはアップグレード

新規インストールやアップグレードを始める前に、使用するプラットフォームの『インストール・ガイド』全体に目を通してください。Adaptive Server の稼動に必要なオペレーティング・システムの条件について検討するには、オペレーティング・システムの管理者に相談することも役立ちます。この稼働条件には、使用するプラットフォームに応じて、メモリ、ロー・デバイス、非同期 I/O、その他の機能の設定が含まれます。このタスクの多くは、インストールを開始する前に実行する必要があります。

サーバをアップグレードする場合は、始める前に必ず master データベース、ユーザ・データベース、トリガ、システム・プロシージャをはじめとするすべてのデータをオフラインでバックアップします。特に古いバージョンと新しいバージョンの間でダンプ・ファイルの互換性がない場合は、アップグレード後すぐにデータの完全なバックアップを別に作成します。

追加のサードパーティ・ソフトウェアのインストール

Adaptive Server は基本的に、各ハードウェア・プラットフォームで一般的なネットワーク・プロトコルをサポートしています。ネットワークが別のプロトコルもサポートしている場合は、必要なプロトコル・サポートをインストールしてください。

Sybase の *interfaces* ファイルの代わりに、ディレクトリ・サービスを使用してサーバのアドレスや他のネットワークの情報を入手できます。ディレクトリ・サービスは、プラットフォームやサードパーティのベンダによって提供されるものであり、Adaptive Server のインストールとは別に購入してインストールする必要があります。Adaptive Server によって現在サポートされているディレクトリ・サービスのリストについては、使用するプラットフォームの「ディレクトリ・サービス」(17 ページ)と『Adaptive Server Enterprise 設定ガイド』を参照してください。

クライアント接続の設定とテスト

クライアントが正しく接続できるかどうかは、Adaptive Server、クライアント・ソフトウェア、ネットワーク製品の組み合わせに依存します。Adaptive Server とともにインストールされるネットワーク・プロトコルを使用する場合のネットワーク接続のテスト方法については、プラットフォームの『Adaptive Server Enterprise 設定ガイド』を参照してください。他のネットワーク・プロトコルを使用する場合は、そのネットワーク製品に添付されている資料を参照してください。Adaptive Server とクライアントとの接続をテストするには、Sybase のコネクティビティ製品付属の “ping” ユーティリティを使用することもできます。クライアントが Adaptive Server に接続する方法の概要については、「Adaptive Server との接続」(16 ページ)を参照してください。*interfaces* ファイルの名前と内容の詳細については、プラットフォームの『Adaptive Server Enterprise 設定ガイド』を参照してください。

物理リソースの割り付け

物理リソースの割り付けとは、パフォーマンスとリカバリの目標を達成するために必要なメモリ、ディスク領域、ワーカー・プロセス、CPU パワーを Adaptive Server に提供することです。新しくサーバをインストールするときに、システム管理者はリソースの使用方法について決定する必要があります。プラットフォームをアップグレードする場合、またはデータベース・システムの設計を変更する場合は、後からメモリ、ディスク・コントローラ、または CPU を追加することによって Adaptive Server のリソースを再割り付けする必要もあります。Adaptive Server とユーザ・アプリケーションのベンチマーク・テストを早めに実行すれば、パフォーマンスのボトルネックになるハードウェア・リソースの不足を特定するのに役立ちます。

Adaptive Server が必要とするディスク リソースの種類を理解するには、『システム管理ガイド 第 2 巻』の「第 16 章 ディスク・リソースの概要」を参照してください。メモリと CPU のリソースについては、『システム管理ガイド 第 2 巻』の「第 3 章 メモリの設定」と「第 5 章 マルチプロセッサ・サーバの管理」を参照してください。

次の項では、物理リソース要件を調べるために役立つ指標について説明します。

専用サーバと共有サーバ

Adaptive Server のリソース計画作成の最初の手順は、同じマシン上で稼働する他のアプリケーションが必要とするリソースを確認することです。通常は、Adaptive Server 専用のマシンを用意します。専用とは、オペレーティング・システムとネットワーク・ソフトウェアが使用する分を除いたリソースを Adaptive Server が自由に使用できるということです。共有システムの場合は、Adaptive Server のクライアント・プログラムやプリント・サーバといった他のアプリケーションが、Adaptive Server と同じマシン上で稼働します。アプリケーションのタイプと使用パターンは時間の経過とともに変化する可能性があるため、共有システム上で Adaptive Server が使用できるリソースを計算するのは困難です。

Adaptive Server 用のリソースを設定するときに、オペレーティング・システム、クライアント・プログラム、ウィンドウ・システムなどで使用されるリソースについて考慮するのはシステム管理者の責任です。使用可能なリソースだけを使用するように Adaptive Server を設定してください。このようにしないと、サーバのパフォーマンスが低下することや、起動できなくなることがあります。

意思決定支援処理と OLTP アプリケーション

Adaptive Server には、OLTP や意思決定支援処理が行われる環境や負荷が一様でない環境でのパフォーマンスを最適化するための多くの機能があります。ただし、このような機能を最大限に活用するには、あらかじめシステム内のアプリケーションの稼働条件を決定します。

負荷が一様でないシステムの場合は、アプリケーションのタイプごとに最も多く使用すると予想される個々のテーブルのリストを作成しておきます。このリストは、アプリケーションにとって最高のパフォーマンスを達成するために役立ちます。

リソースの使用計画

リソースの使用方法について事前に理解し、計画を立てることは非常に大切です。たとえば、ディスク・リソースの場合、Adaptive Server 用にデバイスを初期化して割り付けた後は、Adaptive Server のデータでそのデバイスを使い切ることがないとわかっていても、そのデバイスを他の目的に使用することはできません。同様に、Adaptive Server は設定されたメモリを自動的に予約しますが、このメモリを他のアプリケーションが使用することはできません。

リソースの使用を計画する際の考慮事項

- リカバリのためには、必ず、データベースのトランザクション・ログをデータとは別の物理デバイスに保管してください。『システム管理ガイド 第 2 巻』の「第 6 章 ユーザ・データベースの作成と管理」を参照してください。
- ミッション・クリティカルなデータを保管するデバイスをミラーリングします。『システム管理ガイド 第 2 巻』の「第 2 章 データベース・デバイスのミラーリング」を参照してください。オペレーティング・システムがディスク・アレイとディスク・ミラーリングをサポートしている場合は、Adaptive Server のデータに対してこれらの機能を使用することも検討します。
- テスト用の Adaptive Server を使用している場合は、データベース・デバイスをロー・デバイスではなくオペレーティング・システム・ファイルとして初期化する方が簡単なことがあります。Adaptive Server のデバイスには、ロー・パーティションと動作確認済みのファイル・システムのどちらも使用できます。
- 設定オプションの変更は、Adaptive Server が物理リソース (特にメモリ) を消費する方法に影響する可能性があります。それぞれのパラメータが使用するメモリの量については、「第 5 章 設定パラメータ」を参照してください。

オペレーティング・システムの設定

Adaptive Server で使用可能なリソースと必要なリソースが確定したら、オペレーティング・システム・レベルで次の物理リソースの設定を行います。

- ロー・パーティションを使用する場合は、Adaptive Server が必要とするサイズにロー・デバイスを初期化します。Adaptive Server 用に初期化したロー・デバイスを、オペレーティング・システム・ファイルの保管などの他の目的で使用することはできません。ロー・デバイスを必要なサイズに初期化して設定するときは、オペレーティング・システム管理者に相談してください。
- ネットワーク接続数を設定します。Adaptive Server が稼働するマシンが、設定した数の接続を実際にサポートできることを確認してください。使用するオペレーティング・システム用のマニュアルを参照してください。

- 使用するオペレーティング・システムとアプリケーションの設定がさらに必要な場合があります。使用しているプラットフォームの『ASE インストール・ガイド』を参照してください。また、アプリケーションを実行するためのオペレーティング・システムの条件については、クライアント・ソフトウェアのマニュアルを参照するか、エンジニアに相談してください。

バックアップとリカバリ

データベースを定期的にバックアップするのは、データベース・システムの整合性を保つために重要なことです。Adaptive Server は、システムのクラッシュ (停電による停止など)、またはサーバの障害からは自動的にリカバリを行いますが、メディア障害によって生じるデータの消失からのリカバリができるのは「システム管理者」だけです。

『システム管理ガイド 第 2 巻』の以下の章では、バックアップとリカバリ計画に関する作成と実行について説明します。

- 「第 11 章 バックアップおよびリカバリ・プランの作成」
- 「第 12 章 ユーザ・データベースのバックアップとリストア」
- 「第 13 章 システム・データベースのリストア」
- 「第 16 章 スレッショルドによる空き領域の管理」

master の最新のバックアップの保持

master データベースのバックアップの作成は、バックアップとリカバリの計画において最も重要な要素です。master データベースには、データベース・システム全体の構造についての詳細な情報が格納されています。master database には、Adaptive Server データベース、デバイス、データベースを構成するデバイス・フラグメントの情報が保存されています。Adaptive Server のリカバリ時にこの情報が必要となるので、常に master データベースの最新のバックアップ・コピーを保持することがきわめて重要です。

master データベースのバックアップを常に最新の状態に保つには、ディスク、記憶領域、データベース、またはセグメントに影響するコマンドや次のような手順を実行するたびにデータベースをバックアップします。

- データベースの作成または削除
- 新しいデータベース・デバイスの初期化
- 新しいダンプ・デバイスの追加
- デバイス・ミラーリングに関するコマンドの使用

- master データベースに保管されているシステム・ストアド・プロシージャの作成または削除
- セグメントの作成、削除、変更
- 新しい Adaptive Server ログインの追加

master をテープ・デバイスにバックアップするには、isql を起動して次のコマンドを入力します。

```
dump database master to "tape_device"
```

ここで、*tape_device* はテープ・デバイスの名前です (たとえば */dev/rmt0*)。

システム・テーブルのオフライン・コピーの保存

master の定期的なバックアップに加えて、sysdatabases、sysdevices、sysusages、sysloginroles、syslogins の各システム・テーブルのオフライン・コピーを保存してください。これは、『ASE ユーティリティ・ガイド』で説明している bcp ユーティリティを使用し、それぞれのシステム・テーブルの内容のハードコピーを保管することによって行います。次の出力を印刷してハードコピーを作成します。

```
select * from sysusages order by vstart
select * from sysdatabases
select * from sysdevices
select * from sysloginroles
select * from syslogins
```

これらのテーブルのコピーが保存されていれば、ハード・ディスクの故障などの障害によってデータベースが使用できなくなった場合でも、『システム管理ガイド 第 2 巻』の「第 13 章 システム・データベースのリストア」で説明している手順を使用してリカバリできます。

「記録の保管」(43 ページ) で説明しているように、ユーザ・オブジェクトのデータ定義言語 (DDL) スクリプトのコピーも保存しておいてください。

バックアップ手順の自動化

自動化したバックアップ手順を作成すると、処理を簡単かつすばやく実行できます。バックアップの自動化は、必要なバックアップ・コマンドを実行するためのオペレーティング・システムのスクリプトまたはユーティリティ (UNIX の cron ユーティリティなど) と同様に簡単に使用できます。スレッショルドを使用してさらに手順を自動化することもできます。これについては、『システム管理ガイド 第 2 巻』の「第 15 章 スレッショルドによる空き領域の管理」を参照してください。

❖ **自動バックアップ手順の作成**

自動化スクリプトの作成に必要なコマンドは使用しているオペレーティング・システムによって異なりますが、スクリプトで実行する基本的な手順は同じです。

1　isql を起動して、たとえばテンポラリ・ファイルのような保管領域にトランザクション・ログをダンプします。

2　ダンプ・ファイル名にダンプの日付、時刻、データベース名が含まれるように名前を変更します。

3　履歴ファイル内に新しいバックアップに関する情報を記録します。

4　ダンプ中に発生したエラーを別のエラー・ファイルに記録します。

5　エラーが発生した場合は、システム管理者に自動的にメールを送ります。

データベースをバックアップする前のデータの一貫性の確認

データベースをバックアップする際は、データに一貫性のある正確なバックアップを作成する必要があります。これは master データベースについては特に必要なことです。内部的にエラーがあるデータベースをバックアップすると、それをリストアしたデータベースでも同じエラーが発生します。

バックアップを実行する前にデータベースにエラーがあるかどうかをチェックするには、dbcc コマンドを使用します。ダンプの前には必ず dbcc を使用してデータベースの整合性を検証してください。dbcc によってエラーが検出された場合は、エラーを修正してからデータベースをダンプします。

時間の経過とともに、dbcc を実行していてほとんどエラーが検出されなかった場合は、データベース破壊の危険性が少ないと考えて、dbcc を実行する回数を減らすことができます。データ消失によって受ける影響が大きい場合は、引き続きデータベースのバックアップのたびに dbcc コマンドを実行します。

注意　パフォーマンスを考慮して、通常は dbcc のチェックをピーク時を避けて行うか、別のサーバで行います。

『システム管理ガイド 第 2 巻』の「第 10 章 データベースの一貫性の検査」を参照してください。

ログ・サイズのモニタ

トランザクション・ログの空きがほとんどないときは、トランザクションをダンプするという標準の方法では領域を再利用できなくなる場合があります。システム管理者はログ・サイズをモニタし、定期的にトランザクション・ログのダンプ (通常のデータベース・ダンプの他に) を実行して、そのような事態を回避してください。スレッショルド・ストアド・プロシージャを設定して、ログが所定の容量に達した場合にシステム管理者が通知を受ける (または、ログをダンプする) ようにします。『システム管理ガイド 第 2 巻』の「第 16 章 スレッショルドによる空き領域の管理」を参照してください。Sybase では、データベースのダンプとロードの時間を短縮するために、データベースの完全ダンプを行う直前にトランザクション・ログのダンプを行うこともおすすめします。

sp_helpsegment を使用して、ログ・セグメント内の領域の使用状況を手作業でモニタできます。詳細については、『システム管理ガイド 第 2 巻』の「第 8 章 セグメントの作成と使用」を参照してください。

継続して実行する管理作業とトラブルシューティング

この項では、スケジュール化した定期的なバックアップに加えて、Adaptive Server が使用されている間にシステム管理者が実行する管理作業について説明します。

Adaptive Server の起動と停止

サーバ・マシンの起動と同時に Adaptive Server の起動が行われるように、多くのシステム管理者はこの手順を自動化しています。このように自動化するには、オペレーティング・システムの起動スクリプトを編集するか、オペレーティング・システムの他の手順を使用して行います。Adaptive Server の起動と停止の方法については、使用するプラットフォームの『Adaptive Server Enterprise 設定ガイド』を参照してください。

エラー・ログの表示と削除

エラー・ログの内容を定期的に調べて、重大なエラーが発生していないかどうかを確認してください。オペレーティング・システムのスクリプトを使用して、特定のメッセージを探すためにエラー・ログをスキャンできます。また、特定のエラーが発生したときにシステム管理者に自動的に通知できます。エラー・ログを定期的に調べると、継続的に発生する同質の問題があるか、または特定のデータベース・デバイスに障害が発生しやすくなっていないかどうかを調べることができます。エラー・メッセージとその重大度レベルについては、「第 11 章 システムの問題の診断」を参照してください。

Adaptive Server を起動するたびに情報メッセージとステータス・メッセージがエラー・ログ・ファイルに追加されるので、時間がたつにつれてエラー・ログ・ファイルのサイズが大きくなります。定期的にログ・ファイルを開いて古い記録を削除することによって、ログ・ファイルを「小さく」することができます。ログ・ファイルを管理しやすいサイズに保つようにすれば、ディスク領域の節約につながりエラーの場所も見つけやすくなります。

記録の保管

Adaptive Server システムについての記録を保管することは、システム管理者の重要な作業のひとつです。変更した内容と発生した問題についての正確な記録は、Sybase 製品の保守契約を結んでいるサポート・センタに連絡する場合や、データベースをリカバリする場合に貴重な参考資料になります。また、システム管理者が不在の間、Adaptive Server システムを管理する担当者のために重要な情報を提供できます。

連絡先の情報

システム・セキュリティ担当者、オペレータ、システム上のデータベース所有者だけでなく、システム管理者の連絡先情報のリストも保管してください。それぞれの役割の副担当者の連絡先も記録しておいてください。問題のレポートと対策の要求が適切な担当者へ届くように、Adaptive Server のすべてのユーザにこの情報を伝えます。

設定情報

データベースとデータベース・オブジェクトの作成、および Adaptive Server の設定にはスクリプト・ファイルを使用するようにし、このスクリプト・ファイルを安全な場所に保管するというのが理想的な方法です。スクリプト・ファイルを保管しておくと、システム障害の場合にもシステム全体を作成し直すことができます。また、スクリプト・ファイルを使用すると、新しいハードウェア・プラットフォームで評価用のデータベース・システムを迅速に再作成できます。サードパーティ・ツールを使用してシステム管理を行っている場合は、管理作業を実行した後で同じスクリプトを生成してください。

次のような情報を記録してください。

- データベースとデータベース・オブジェクトを作成するために使用するコマンド (DDL スクリプト)
- Adaptive Server の新しいログインとデータベース・ユーザを追加するコマンド

- 現在の Adaptive Server の設定ファイル (「設定ファイルを指定して sp_configure を使用する方法」(64 ページ) を参照)
- データベース・デバイスとして初期化されたすべてのファイルとロー・デバイスの名前、ロケーション、サイズ

Adaptive Server の設定に対するすべての変更について、日付入りのログを保持してください。結果の要約だけでなく、変更の理由と日時を示す簡単な説明をそれぞれの変更に付記してください。

管理作業のスケジュール

スケジュール化した定期的な管理作業用のカレンダを作成し、自分のサイトで実行する手順をそのカレンダに記入します。たとえば、次のようなものがあります。

- dbcc を使用して行うデータベースの一貫性チェック
- ユーザ・データベースとシステム・データベースのバックアップ
- トランザクション・ログの空き領域のモニタ (自動的に行われない場合)
- トランザクション・ログのダンプ
- Adaptive Server、Backup Server、Adaptive Server Monitor のエラー・ログの内容の調査
- update statistics コマンドの実行 (『パフォーマンス&チューニング・シリーズ : 統計分析によるパフォーマンスの向上』の「第 1 章 set statistics コマンドの使用」を参照)
- 監査情報の調査 (監査オプションをインストールしている場合)
- ストアド・プロシージャの再コンパイル
- サーバ・マシンのリソース使用状況のモニタ

システム情報

Adaptive Server を実行するハードウェアとオペレーティング・システムに関する次の情報も記録してください。

- オペレーティング・システムの設定ファイルまたは起動ファイルのコピー
- ネットワーク設定ファイル (たとえば、*hosts* ファイルと *services* ファイル) のコピー
- Adaptive Server の実行ファイルとデータベース・デバイスの名前とパーミッション

- バックアップに使用するテープ・デバイスの名前とロケーション
- 自動バックアップ、Adaptive Server の起動、その他のシステム管理アクティビティを行うためのオペレーティング・システム・スクリプトまたはプログラムのコピー

災害時のリカバリ計画

基本的なバックアップとリカバリの手順、「バックアップとリカバリ」(39 ページ) のガイドライン、データのリカバリについての自分自身の経験をまとめて、システムに合ったリカバリ手順を簡単なリスト形式で作成してください。このリストは、自分にとっても、緊急時に運用システムをリカバリする必要がある他のシステム管理者にも役立ちます。

その他のリソース

システム管理者が覚える情報は非常に多くありますが、基本的な管理作業の習得と簡素化に役立つソフトウェア・ツールがいくつか提供されています。これらのソフトウェア・ツールには、サーバのパフォーマンスやその他のアクティビティをモニタする Adaptive Server Monitor や、多くの管理作業を簡単に実行できる Sybase Central などがあります。また、システム管理者が行う日常の管理作業を支援するサードパーティ製のソフトウェア・パッケージもあります。

第 4 章

Adaptive Server Plug-in for Sybase Central の概要

この章では、Sybase Central を使用して Adaptive Server を管理する方法について説明します。この章は、Sybase Central の概要を説明することを目的としています。Adaptive Server プラグインの機能の詳細については、Sybase Central のオンライン・ヘルプを参照してください。

Adaptive Server Sybase Central Plug-in の概要

Sybase Central は、グラフィカル・ユーザ・インタフェース (GUI) 管理ツールです。Sybase Central では、特定の Sybase 製品を管理する各種「プラグイン」を使用できます。Adaptive Server プラグインを使用して Adaptive Server を管理することにより、Transact-SQL コマンドやシステム・ストアド・プロシージャの構文を覚えなくても複雑な管理作業を実行できます。Adaptive Server プラグインを使用して、次のような操作を実行できます。

- 1 台のコンソールから複数のサーバを管理する － Sybase Central メイン・ウィンドウから、すべての Adaptive Server インストールを管理できます。
- データベース定義言語 (DDL) を作成する － Adaptive Server でオブジェクトの DDL を作成できます。
- オブジェクトを視覚的に表示する － 各 Adaptive Server のデータベースとログイン、および各データベース内のオブジェクトを表示できます。ウィンドウを閉じたり開いたりして、データベースやログインの情報を表示できます。Adaptive Server プラグインでは、次のように多数の項目に関する情報が表示されます。
  - データベースとテーブル
  - ディスク・デバイス

  - アクティブなプロセスとロック
  - ログインとユーザ
  - データ・キャッシュ
  - ASE Replicator、Job Scheduler、メッセージング・サービス
  - Interactive SQL などの他のユーティリティへのアクセス (クエリの送信とクエリ結果の表示)
- 関連オブジェクト間のナビゲーション － 表示中のプロパティ・シートのオブジェクトと関連があるデータベース・オブジェクトの詳細を取得するには、表示されたオブジェクトのダイアログ・ボックスを通じて関連オブジェクトに直接ナビゲートします。
- クラスタの作成 － Adaptive Server Cluster Edition を購入した場合は、Adaptive Server プラグインを使用してクラスタを作成できます。オンライン・ヘルプと『Cluster ユーザーズ・ガイド』を参照してください。

Adaptive Server プラグインとコマンド・ラインの更新

Adaptive Server plug-in for Sybase Central は、さまざまな Adaptive Server Enterprise 製品を管理します。15.0.3 より前のバージョンでは、Adaptive Server プラグインは Sybase Central 4.3 上で実行されます。15.0.3 では、Adaptive Server プラグインは Sybase Central 6.00 上で実行されます。15.0.3、Sybase Central 6.00 の新機能を次に示します。

- 検索ツールは、プラグインによって表示されるオブジェクトを見つけるのに役立ちます。[表示] - [検索ウインドウ枠] を選択し、オブジェクトが属するプラグインに応じてオブジェクトを選択します。
- [接続プロファイル]、[インポート]、[エクスポート] オプションを使用すると、プロファイル接続にテキストの説明を追加できます。接続プロファイルをファイルにインポートしたり、ファイルからエクスポートすることもできます。これにより、ユーザ間で接続プロファイルを共有できます。
- Windows Vista に対する優れたサポート

バージョン 15.0.3 Adaptive Server プラグインの新機能を次に示します。

- [Sybase Central] コンテキスト・バーの下 (標準ツールバーの下) にあるコンテキストで区別されるツールバーから [追加] アイコンを選択してオブジェクトを作成する。
- ストアド・プロシージャと SQLJ プロシージャは、[プロシージャ] フォルダに格納される。

- スカラ関数またはユーザ定義の関数がサポートされるようになり、これらは [Functions] フォルダ内に SQLJ とともに格納される。これらの関数に関する説明は、Adaptive Server プラグインのヘルプにも記載されています。
- ユーティリティ項目に、標準ツールバーの下にあるコンテキストで区別されるツールバー (コンテキスト・バー) のメニューからアクセスが可能。

Adaptive Server プラグインに付属していた DBISQL11 は、別個の製品 (バージョン 11.0) となる。次の拡張機能が含まれます。

- 複数の結果セットの数が 10 に制限されなくなる。
- Adaptive Server のログイン・ダイアログは保持され、最後の 5 回の接続サーバ名を表示する。
- DBISQL11 または対話型 SQL で接続プロファイルと同様の機能、接続のお気に入り機能がサポートされるようになる。
- [SQL 文] ウィンドウ枠内に行番号が含まれるようになる。
- [結果] ウィンドウ枠には、選択した行から、select all, insert/update/delete SQL 文の使用、並べ替え、生成を行った結果が表示される。

Adaptive Server プラグインの使用

Adaptive Server Plug-in for Sybase Central では、直感的でわかりやすい操作によって Adaptive Server Enterprise を管理できます。Sybase Central では、左側のウィンドウ枠に Adaptive Server プラグインが表示されます。このウィンドウ枠に、プラグインで管理できるさまざまなオブジェクトのフォルダが階層形式でリストされます。次のような作業を実行できます。

- オブジェクトの特性の表示および変更
- 別のオブジェクトの作成
- オブジェクトを作成するための SQL テキストの作成 (Adaptive Server オブジェクトのリバース・エンジニアリング)
- オブジェクトの削除
- Adaptive Server の設定
- 以下の項目の管理：
  - データベース・デバイス
  - プロキシ・データベースとテンポラリ・データベース
  - インデックス
  - パーティション

  - セグメント
  - トリガ
  - ログインと役割
  - ビュー
  - ASE Replicator
- Job Scheduler での Adaptive Server ジョブの設定
- Adaptive Server の起動と停止
- クエリの実行
- ユーザの操作に基づいた、プラグインで作成した SQL 文のログ

Sybase Central の起動と停止

Sybase Central を起動するには、次の手順に従います。

- UNIX の場合、*$SYBASE/shared/sybcentral600* ディレクトリに移動し、*scjview.sh* スクリプトを実行します。
- Windows の場合、[スタート] から、[プログラム] - [Sybase] - [Sybase Central v6] を選択します。

  Windows の場合、*%SYBASE%¥Shared¥Sybase Central 6.0.0¥* ディレクトリに移動し、*scjview.bat* スクリプトを実行します。

Sybase Central を終了するには、[ファイル] - [終了] を選択します。

Adaptive Server プラグインの登録

Adaptive Server プラグインは、サーバのインストール処理の一部として Sybase Central に登録されます。ただし、Adaptive Server プラグインが適切に登録されない場合は、Adaptive Server プラグインを手動で登録できます。

- Unix の場合、*$SYBASE/ASEP/bin/registerASEP* を実行します。
- Windows の場合、*%SYBASE%¥ASEP¥bin¥registerASEP.bat* を実行します。
- Adaptive Server プラグインを手動で登録するには、次の手順に従います。

  a [ツール] - [プラグイン] - [登録] を選択します。登録ウィザードが表示されます。

  b [登録] を選択します。

c　[プラグイン登録ファイルの指定によって、プラグインを登録します。] を選択します。

d　[参照] をクリックします。

e　*$SYBASE/ASEP/bin* (Windows の場合は *%SYBASE%¥ASEP¥bin*) に移動し、*ASEPlugin.jpr* を選択します。ウィザードの手順に従って Adaptive Server プラグインを登録します。Adaptive Server プラグインを使用して、次のような操作を実行できます。

共通の作業の実行

Adaptive Server プラグインでユーザがよく実行する作業を以下に示します。

各作業の詳細については、Adaptive Server プラグインのオンライン・ヘルプを参照してください。

Adaptive Server の起動と停止

Unified Agent で Adaptive Server をモニタしている場合は、サーバを右クリックし、[停止]、[起動]、または [再起動] を選択することによって、サーバを起動、停止、再起動します。

Adaptive Server を Unified Agent でモニタしていない場合は、[停止] を選択してサーバを停止します。

Adaptive Server との接続

Adaptive Server に接続するには、次のいずれかの方法を使用します。

- ツールバーの接続アイコンを選択します。
- [Adaptive Server Enterprise] を右クリックし、メニューから [接続] を選択します。
- サーバ・グループを右クリックし、メニューから [接続] を選択します。

  [Adaptive Server Enterprise] フォルダまたは接続アイコンから接続を開始した場合は、接続されたサーバが 「デフォルト」サーバ・グループに表示されます。サーバ・グループから接続を開始した場合は、該当するサーバ・グループに “Connected to server” と表示されます。

接続するサーバを指定するには、次のいずれかの方法を使用します。

- [接続] ダイアログ・ボックスで、サーバのホスト名とポート番号を指定します。
- [サーバ名] ドロップダウン・リストから、事前に定義された Adaptive Server を選択します。このドロップダウン・リストは、interfaces ファイル (UNIX)、*sql.ini* ファイル (Windows)、LDAP サーバのサーバ・リストで構成されます。
- [接続] ダイアログで [検索] をクリックして、使用可能な Adaptive Server を検出します。この方法を使用するには、Adaptive Server Enterprise のプロパティ・ページにある [Server Discovery] タブで、検出サーバをあらかじめ定義しておく必要があります。

データベースの作成

データベースを作成する前に、使用する予定のデータベース・デバイスに使用可能な領域が十分あることを確認してください。

データベースを作成するには、次の手順に従います。

右ウィンドウ枠で、[データベースの追加] アイコンを右クリックします。または、次の手順を実行します。

1 [データベース] フォルダを選択します。

2 [ファイル] - [新規] - [データベース] を選択するか、[データベース] フォルダで右クリックし、[新規] - [データベース] を選択します。[データベースの追加] ウィザードが開きます。[データベースの追加] ウィザードでは、以下の情報が要求されます。

表 *4-1:* [データベースの追加] ウィザードの入力項目

入力項目	説明
データベース名	データベースの名前を入力する。
データベース・デバイス	新しいデータベースを割り付けるデータベース・デバイスを指定する。
データベース・デバイスのサイズ	各データベース・デバイスのサイズを指定する。
データまたはログ	データベース・デバイスにデータとトランザクション・ログのどちらを保存するかを指定する。
上書き	同じデバイスにデータとログを保存する場合は、上書きを指定する。
ロード用	バックアップからリストアできるようにデータベースを作成する場合は、[ロード用] チェック・ボックスをオンにする。これは、メディア障害からリカバリする場合、またはある場所から別の場所にデータベースを移動する場合のみに適用される。
guest アカウント	データベースに guest ユーザを作成するかどうかを指定する。

サイズを入力しなかった場合は、database size 設定変数の値と *model* データベースのサイズのうち、大きい方が割り当てられます。

記憶領域が限られており、かつ、トランザクション・ログとデータを同じ論理デバイスに格納しなければならない場合は、[上書き] をオンにすると、データと別のデバイス・フラグメントでログを管理できます。

データベースの作成後は、データベースを削除しない限り、データベース・デバイスの削除や変更を行うことはできません。

警告！ データベースを削除すると、そのデータベースのすべてのオブジェクトが削除されます。

データベースの削除

データベースを削除できるのは、そのデータベースの所有者のみです。

データベースを削除するには、次の手順に従います。

1 データベース・アイコンを選択します。

2　[編集] - [削除] を選択します。

3　[削除の確認] ダイアログ・ボックスで削除を確認します。

注意　ユーザ・データベースを削除した後、master データベースをバックアップすることをおすすめします。

ユーザの追加

データベース所有者は、自分が所有するデータベースでユーザを追加および削除できます。

ユーザを作成するには、次の手順に従います。

1　[データベース] フォルダを展開し ([+] アイコンを選択)、[ユーザ] フォルダを選択します。

2　[ファイル] - [新規] - [ユーザ] を選択します。

[ユーザの追加] ウィザードが起動し、以下の情報を要求します。

表 *4-2:* [ユーザの追加] ウィザードの入力項目

入力項目	説明
名前	ユーザの名前。この名前はログイン名と同じである必要はない。
ログイン名	このユーザが割り当てられるログイン。
グループ	ユーザにグループを割り当てる (任意選択)。デフォルト：public

注意　ユーザをいずれかのグループに割り当てることができます。どのグループにも割り当てられていないユーザは、デフォルトの “public” グループのメンバになります。

または、[ユーザ] フォルダを選択します。[ユーザ] フォルダで右クリックし、[新規] - [ユーザ] を選択します。

ユーザの削除

オブジェクトを所有しているユーザは削除できません。オブジェクトの所有権を譲渡するコマンドはないので、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。同様に、他のユーザにパーミッションを付与しているユーザを削除するには、最初にパーミッションをカスケード付きで取り消す必要があります。必要に応じて、他のユーザに再度パーミッションを付与します。

ログインのロックは、ユーザ削除の簡単な代替方法です。

ユーザを削除するには、次の手順に従います。

1　ユーザ・アイコンを選択します。

2　[編集] - [削除] を選択します。

3　[削除の確認] ダイアログ・ボックスで削除を確認します。

または、ユーザ・アイコンを右クリックして [ユーザ] フォルダを選択し、[削除] を選択することもできます。

ユーザを削除する前に、次の操作を行ってください。

1 ユーザのコマンドとオブジェクトのパーミッションをカスケード付きで取り消します。

2 必要に応じて、他のユーザに再度パーミッションを付与します。

3 ユーザのオブジェクトを削除します。

テーブルの作成

create table パーミッションを持つデータベース所有者またはユーザだけがテーブルを作成できます。

テーブルを作成するには、次の手順に従います。

1 作業しているデータベースで [ユーザ・テーブル] フォルダを選択します。

2 [ファイル] - [新規] - [テーブル] を選択するか、[ユーザ・テーブル] フォルダで右クリックし、[新規] - [ユーザ] を選択します。

テーブル • エディタが開きます。

3 [名前] ボックスに名前を入力します。

4 [所有者] リストから所有者を選択します。デフォルトは “dbo” です。

または、[ユーザ・テーブル] フォルダを選択します。右ウィンドウ枠で、[テーブルの追加] アイコンをダブルクリックします。

テーブルの削除

テーブルの削除は、そのテーブルを参照している他のオブジェクトがないことを確認してから行ってください。参照しているオブジェクトがある場合は、それらのオブジェクトを編集してエラーを回避します。他のオブジェクトがテーブルを参照しているかどうかを確認するには、その依存性を調べます。

注意 テーブルを削除すると、そのテーブルに関連付けられているインデックスとトリガが削除され、そのカラムにバインドされているルールまたはデフォルトがバインド解除されます。

テーブルを削除できるのは、テーブルの所有者だけです。

テーブルを削除するには、次の手順に従います。

- 次の手順に従ってください。
  - テーブル・アイコンを選択します。
  - [編集] - [削除] を選択します。
  - [削除の確認] ダイアログ・ボックスで削除を確認します。
- または、テーブル・アイコンを右クリックしてテーブルを選択し、[削除] を選択することもできます。

サーバ・グループの作成

サーバ・グループを作成するには、次の手順に従います。

1　[Adaptive Server Enterprise] を選択します。

2　[ファイル] - [新規] - [サーバ・グループ] を選択します。

3　[サーバグループの追加] ウィザードの手順に従います。

または、右ウィンドウ枠で [新規] - [サーバ・グループ] を選択してサーバ・グループを追加することもできます。

サーバ・ステータスの取得

Adaptive Server を Unified Agent でモニタしている場合は、次のいずれかの方法でサーバのステータスを確認します。

- サーバが所属するサーバ・グループをクリックします。サーバ・グループの [詳細] ウィンドウ枠の [ステータス] カラムを確認します。
- [Sybase Central] の下の [Adaptive Server Enterprise] をクリックし、右ウィンドウ枠の [サーバ] タブをクリックします。サーバのステータスは [ステータス] カラムに表示されます。
- サーバ・アイコンの右下に緑色の三角形が表示されている場合、Adaptive Server は稼動中です。赤い正方形は、Adaptive Server が停止中であることを示します。

注意 デフォルトでは、Adaptive Server プラグインのサーバ・ステータスのチェックは無効になっています。Unified Agent で Adaptive Server をモニタできるようにするには、次の手順に従います。

- [Adaptive Server Enterprise] を右クリックし、[プロパティ] を選択します。
- [ユーザ独自の設定] を選択し、[Unified Agent (UA) 関連の機能を有効にする] を選択します。

サーバ・ログの取得

Unified Agent で Adaptive Server をモニタしている場合は、サーバを選択し、右ウィンドウ枠の [Server Log] タブをクリックして、サーバ・ログを取得します。

サーバ・ログは、サーバ・ログのフィルタ設定に基づいて取得されます。サーバ・ログのフィルタを設定するには、サーバを右クリックし、[サーバ・ログ・フィルタ] を選択します。デフォルトでは、最新の 1000 行をサーバ・ログから取得します。サーバのフィルタ設定に基づいて、次のサーバ・ログを取得できます。

- ログ・ファイル全体
- 最新の *n* 行
- 最新の *n* 日間のログ
- 正規表現に一致する行

SQL 文のロギング

Adaptive Server プラグインによって実行されたすべての SQL 文のログを取るには、次の手順に従います。

- サーバを右クリックし、[SQL 文のログを取る] を選択します。
- SQL 文のログをウィンドウに直接送るか、ファイルに送るかを選択します。

SQL 文の実行

Interactive SQL クエリ・ツールを使用して、Adaptive Server プラグイン内から SQL 文を実行します。Interactive SQL ツールを起動するには、次のいずれかの手順に従います。

SQL 文を実行するサーバを右クリックし、メニューから [対話型 SQL のオープン] を選択します。または、以下の手順に従います。

1 [Adaptive Server Enterprise] をクリックします。

2 右ウィンドウ枠の [ユーティリティ] タブをクリックし、[Interactive SQL] を選択します。

サーバ・グループに所属するサーバのセットに対して SQL 文を同時に実行します。次の手順に従います。

1 サーバ・グループを右クリックし、[SQL の実行] を選択します。

2 SQL 文を実行するサーバを選択します。

3 [実行] をクリックします。

各サーバの結果セットは、[SQL Execution] ダイアログの [結果セット] ウィンドウ枠に表示されます。

SQL の実行プランとコスト情報の表示

Adaptive Server プラグインを使用して、各クエリの SQL 実行プランの GUI バージョン (showplan の GUI バージョン)、およびストアド・プロシージャのすべてのクエリの実行プランの GUI バージョンを表示します。この GUI 表示では、実行プランの演算子ごとにノードが含まれます。

GUI でプランを取得するには、次の手順に従います。

1 Interactive SQL を起動します。

2 クエリまたはストアド・プロシージャを実行します。

3 [ツール] - [プランビューワを開く] を選択します。

4 SQL ウィンドウ枠下のドロップダウン・リストからクエリを選択します。

5 [詳細] タブをクリックして、選択したクエリの GUI プランを表示します。演算子ノードをクリックして、ノードの統計情報の詳細を表示します。

6 [XML] タブをクリックして、選択したクエリの実行プランを XML 形式で表示します。

7 [Text] タブをクリックして、送信したクエリの実行プランをテキスト形式で表示します。

Interactive SQL の詳細については、「Interactive SQL の起動」(58 ページ) を参照してください。

オブジェクト・プロパティの表示と更新

[プロパティ] ダイアログを使用して、Adaptive Server プラグインで表示されるすべてのオブジェクトの設定を表示および修正します。

[プロパティ] ダイアログを開くには、次の手順に従います。

1　表示または修正するオブジェクトをクリックします。

2　オブジェクトを右クリックし、[プロパティ] を選択します。

3　実行する作業に合った適切なタブを選択します。

4　[プロパティ] ダイアログで、必要な修正を加えます。

5　[適用]、[OK]、または [キャンセル] をクリックします。

オブジェクトを作成するための SQL テキストの生成

オブジェクトの作成に必要な SQL テキストを生成します。これは、オブジェクトのリバース・エンジニアリングが可能であることを意味します。SQL テキストを作成するには、オブジェクトを右クリックし、[DDL の生成] を選択します。

Adaptive Server 設定パラメータの表示と更新

[サーバのプロパティ] ダイアログで、Adaptive Server 設定パラメータを表示および更新します。

1　サーバを右クリックし、メニューから [設定] を選択します。

2　[設定パラメータを表示] のドロップダウン・リストで、機能グループを選択します。

3　表示または更新するパラメータを見付けて選択します。

4　更新する必要がある場合は、[値] カラムに、新しい値を入力します。

5　必要に応じて、[適用]、[OK]、または [キャンセル] をクリックします。

Interactive SQL の使用

Interactive SQL を使用して、SQL 文の実行、スクリプトの作成、サーバへのデータベース・データの表示ができます。Interactive SQL は、次の目的で使用できます。

- データベース内の情報をブラウズする。
- アプリケーションで使用する予定の SQL 文をテストする。
- クエリ結果をファイルに保存する。
- 結果セットのデータを編集する。
- データをデータベースにロードし、管理作業を実行する。

Interactive SQL では、コマンド・ファイルまたはスクリプト・ファイルを実行することもできます。たとえば、データベースに実行する繰り返し可能なスクリプトを作成し、Interactive SQL を使用してそれらのスクリプトをバッチで実行できます。

Interactive SQL の起動

Sybase Central からの Interactive SQL の起動

Interactive SQL を起動するには、次のいずれかの手順を実行します。

- Sybase Central でデータベースを選択し、[ファイル] - [対話型 SQL のオープン] を選択します。Interactive SQL がデータベースに接続します。または、データベースを右クリックし、[対話型 SQL のオープン] を選択することもできます。

  メニュー項目 [対話型 SQL のオープン] は、サーバへの接続をオープンします。ただし、サーバのメニュー項目を選択した場合は、そのサーバのデフォルト・データベースへの接続をオープンします。[対話型 SQL のオープン] メニューから特定のデータベースを選択すると、選択されたデータベースに対して Interactive SQL がオープンします。

- サーバに接続しないで Interactive SQL を起動するには、[ツール] - [Adaptive Server Enterprise] - [対話型 SQL] を選択します。[接続] ダイアログが表示されます。

コマンド・ラインからの Interactive SQL の起動

コマンド・ラインから Interactive SQL を起動する手順は、使用するオペレーティング・システムによって異なります。

Interactive SQL を単独で起動すると、[接続] ダイアログが表示され、Sybase Central の場合と同じ方法でデータベースに接続できます。

- UNIX の場合は、*$SYBROOT/DBISQL/bin* ディレクトリに移動し、次のように入力します。

  ```
  dbisql
  ```

  Windows の場合は、*%SYBROOT%¥DBISQL¥bin* ディレクトリに移動し、次のように入力します。

  ```
  dbisql.bat
  ```

- [接続] ダイアログで、[接続] ダイアログ・ボックスにデータベースへの接続情報を入力し、[OK] をクリックします。

新しい Interactive SQL ウィンドウを開くには、次の手順に従います。

1 [ウインドウ] - [新しいウインドウ] を選択します。[接続] ダイアログが表示されます。

2 [接続] ダイアログで接続オプションを入力し、[OK] をクリックして接続します。

  接続情報 (データベース名、ユーザ ID、データベース・サーバなど) は、[SQL 文] ウィンドウ枠の上のタイトル・バーに表示されます。

また、SQL メニューの [接続] コマンドと [切断] コマンドを使用するか、[SQL 文] ウィンドウ枠で connect または disconnect 文を実行することにより、データベースへの接続やデータベースからの切断を行うこともできます。

第 5 章

設定パラメータ

この章では、Adaptive Server の設定パラメータについて説明します。パラメータはアルファベット順に並べてあります。

設定パラメータは、sp_configure を使用してユーザが設定できるパラメータです。設定パラメータは、基本的なサーバ操作から特殊なサーバ操作に至る広範囲なサービスと、パフォーマンス・チューニングに使用します。

概要

設定パラメータは、Adaptive Server の動作に関するさまざまな制御を行うためにユーザが設定できるパラメータです。すべての設定パラメータにデフォルト値が用意されています。設定パラメータを使用して、インストール環境の特定のニーズを満たすように Adaptive Server をカスタマイズします。

この章を参照してサーバのパフォーマンスを最適化するためにどの設定パラメータを設定し直したらよいか、慎重に調べてください。

警告！ 設定パラメータの変更は、十分注意して行ってください。パラメータ値を不用意に変更すると、Adaptive Server のパフォーマンスやサーバ・オペレーションの他の部分に悪影響を与えることがあります。

Adaptive Server の設定ファイル

Adaptive Server の設定パラメータの値は、ASCII テキスト・ファイルである設定ファイルに保存されます。新しく Adaptive Server をインストールするときは、パラメータはデフォルト値に設定されます。デフォルトでは、この設定ファイルの名前は *server_name.cfg* で、Sybase Adaptive Server のホーム・ディレクトリ ($SYBASE_ASE) に保存されます。設定パラメータを変更するたびに、それまでの設定ファイルのコピーが作成されますが、その名前は *server_name.001*、*server_name.002*、*server_name.003*、...、*server_name.999* というように付けられます。新しい値は、*server_name.cfg* ファイルまたは起動時に指定したファイルに書き込まれます。

設定パラメータの変更

設定パラメータを設定または変更するには、次の 3 つの方法があります。

- sp_configure に該当するパラメータと値を指定して実行する。
- 設定ファイルを編集してから、configuration file オプションを指定して sp_configure を実行する。
- 起動時に設定ファイル名を指定する。

設定パラメータには、動的パラメータと静的パラメータがあります。動的パラメータは、sp_configure を実行するとすぐに有効になります。静的パラメータの場合は、メモリを再割り付けする必要があるので、Adaptive Server を再起動しないと有効にはなりません。この章の各パラメータの説明には、そのパラメータが静的か動的かが記載されています。

ユーザが値を変更すると、新しい値がシステム・テーブル sysconfigures と設定ファイルに書き込まれます。現在の設定ファイルと sysconfigures は、実行値ではなく、設定値を反映しています。システム・テーブル syscurconfigs は、設定パラメータの現在の実行値を反映しています。

設定パラメータの変更に必要な役割

sp_configure を使用するために必要な役割は次のとおりです。

- どのユーザも sp_configure を実行して、パラメータおよびパラメータの現在の値を表示できます。
- sp_configure を実行して設定パラメータを変更できるのは、システム管理者とシステム・セキュリティ担当者だけです。
- sp_configure を実行して次の設定パラメータの値を変更できるのは、システム・セキュリティ担当者だけです。
  - allow procedure grouping
  - allow remote access

- allow sendmsg
- allow updates to system tables
- auditing
- audit queue size
- check password for digit
- current audit table
- enable ldap user auth
- enable pam user auth
- enable ssl
- log audit logon failure
- log audit logon success
- maximum failed logins
- minimum password length
- msg confidentiality reqd
- msg integrity reqd
- secure default login
- select on syscomments.text
- SQL Perfmon Integration
- syb_sendmsg port number
- suspended audit when device full
- systemwide password expiration
- unified login required
- use security services

sp_configure による単位の指定

sp_configure では、設定パラメータの値の単位を単位指定子で指定できます。単位指定子には、ページ数を表す p または P、メガバイト数を表す m または M、ギガバイト数を表す g または G があります。メモリの量を指定するパラメータを設定するときに単位を指定しなかった場合は、論理ページ・サイズが基本単位として使用されます。

注意 メモリに関係するパラメータを設定するときは、P (ページ・サイズ) パラメータ以外の単位指定は使用しないでください。メモリに関係するパラメータの設定時に他のパラメータを使用すると、算術オーバフローのエラー・メッセージが返されることがあります。

単位指定の構文は次のとおりです。

```
sp_configure "parameter name", 0, "p|P|k|K|m|M|g|G"
```

“0” をプレースホルダとして必ず指定します。

この単位指定はすべてのパラメータで使用できます。たとえば、number of locks を 1024 に設定する場合は、次のように入力します。

```
sp_configure "number of locks", 1024
```

または

```
sp_configure "number of locks", 0, "1K"
```

この機能を使用しても、sp_configure の出力の内容は変わりません。

設定パラメータのヘルプ情報の取得

特定の設定パラメータのヘルプ情報を参照するには、sp_helpconfig または sp_configure を使用します。次に例を示します。

```
sp_helpconfig "number of open"

Configuration option is not unique.
option_name                    config_value  run_value
------------------------------ ------------ -----------
number of open databases                 12          12
number of open indexes                  500         500
number of open objects                  500         500


                    sp_helpconfig "number of open indexes"
```

```
number of open indexes sets the maximum number of indexes that can be open at
one time on SQL Server. The default value is 500.
Minimum Value Maximum Value Default Value Current Value Memory Used
------------- ------------- ------------- ------------- -----------
          100    2147483647           500           500         208


sp_configure "number of open indexes"

Parameter Name           Default  Memory Used  Config Value  Run Value
-----------------------  -------  -----------  ------------  ---------
number of open indexes       500          208           500        500
```

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

sp_configure の使用

sp_configure は、設定パラメータを表示または再設定します。sp_displaylevel を使用して次のレベルのいずれかに表示レベルを設定することによって、sp_configure が表示するパラメータの数を制限できます。

- 基本
- 中間
- 包括

表示レベルの詳細については、「パラメータ階層のユーザ定義サブセット：表示レベル」(71 ページ) を参照してください。sp_displaylevel 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。

表 5-1 は、sp_configure の構文の説明です。「結果」の欄は、表示レベルを「包括」に設定した場合についての説明です。

表 *5-1: sp_configure* の構文

コマンド	結果
sp_configure	すべての設定パラメータについて、現在の値、デフォルト値、前回の設定値、設定に必要なメモリ量をグループごとに表示する。
sp_configure “*parameter*”	現在の値、デフォルト値、前回の設定値、指定されたパラメータが使用するメモリ量を表示する。
sp_configure “*parameter*”, *value*	*parameter* の値を *value* に再設定する。
sp_configure “*parameter*”, 0, “default”	指定されたパラメータをデフォルト値にリセットする。
sp_configure “*group_name*”	*group_name* 内のすべての設定パラメータについて、現在の値、デフォルト値、前回の設定値、設定に必要なメモリ量を表示する。
sp_configure “configuration file”, 0, “*sub_command*”, “*file_name*”	設定ファイルから設定パラメータを設定する。パラメータの詳細については、「設定ファイルを指定して sp_configure を使用する方法」(64 ページ) を参照してください。

構文の要素

表 5-1 のコマンドで使用している要素は次のとおりです。

- *parameter* は、有効な Adaptive Server 設定パラメータまたはパラメータの部分文字列です。
- *value* は、そのパラメータに有効な範囲内の整数値です。有効な範囲については、それぞれのパラメータの説明を参照してください。トグル式のパラメータに有効な値は、1 (オン) と 0 (オフ) だけです。1 (オン)、0 (オフ)。
- *group_name* は、パラメータ階層内のグループの名前です。

パラメータの解析

sp_configure は、それぞれのパラメータ (およびパラメータ名の一部) を“%parameter%”として解析します。文字列によって特定されるパラメータが 1 つでない場合は、その文字列に一致するすべてのパラメータの値を返します。

次の例では、lock shared memory、number of locks、lock promotion HWM、server clock tick length、print deadlock information、deadlock retries などの“lock”を含むすべての設定パラメータの値が返されます。

```
sp_configure "lock"
```

注意 sp_configure でパラメータ名の一部を指定してパラメータ値を設定するときに、一致するパラメータが 2 つ以上ある場合は、そのパラメータ名部分に一致するすべてのパラメータの現在値が返され、ユニークなパラメータ名の入力を要求するプロンプトが表示されます。

設定ファイルを指定して *sp_configure* を使用する方法

Adaptive Server の設定は、上記のように sp_configure を使用して対話型で行うことも、編集またはリストアした設定ファイルから値を読み込む非対話型で行うこともできます。

設定ファイルから変更することで、以下のことを行えます。

- 同一の設定ファイルを使用することによって、複数のサーバに特定の設定を複写する
- 各自のサーバ上で設定値をテストするための基準として、設定ファイルを使用する
- 実際に値を設定する前に、パラメータ値を検証するために設定ファイルを使用する
- 複数の設定ファイルを作成し、リソースの変化に応じて設定ファイルを切り替える

ファイルの編集方法については、「設定ファイルの編集」(66 ページ) を参照してください。起動時に設定ファイル名を指定する方法については、「設定ファイルを指定した Adaptive Server の起動」(68 ページ) を参照してください。

設定ファイルの名前付けについてのヒント

設定ファイルの名前をデフォルトの名前から変更するときに、ファイル名の *server_name* 部分をそのまま保持する場合は、必ず拡張子に英字を 1 つ以上入れてください (例：*my_server.A001*)。または、ファイル名の *server_name* 部分を変更することもできます (例：*A_my_server.001*)。このようにすれば、パラメータを修正するときに自動的に生成されるバックアップ設定ファイルとの混同を避けることができます。

sp_configure を使用した設定ファイルの読み込みまたは書き込み

sp_configure で configuration file オプションを使用する場合の構文は、次のとおりです。

```
sp_configure "configuration file", 0, "subcommand", "file_name"
```

各パラメータの意味は、次のとおりです。

- “configuration file” (引用符も必要) － このコマンドで設定ファイルを使用することを指定する。
- 0 － configuration file パラメータの後に必要 (下位互換性のため)。
- “*subcommand*” － 以下のいずれかを指定する。
  - write は、現在の設定を使用してファイル *file_name* を作成します。*file_name* が既に存在する場合は、エラー・ログにメッセージが出力され、既存ファイルの名前は *server_name.001*、*server_name.002*、... という命名規則に従って変更されます。静的パラメータを変更した後にサーバを再起動しないで write を使用した場合は、そのパラメータの現在実行している値が表示されます。*file_name* のディレクトリを指定しない場合は、ファイルは Adaptive Server が起動されたディレクトリに書き込まれます。
  - read は、*file_name* 内に記述されている値を検証し、検証をパスした値をサーバに読み込みます。*file_name* に記述されていないパラメータには、そのパラメータの現在の値が使用されます。

    file_name 内の静的パラメータの値が現在の実行値と異なる場合、read は失敗し、メッセージが表示されます。ただし、この場合も *file_name* 内の値の検証は実行されます。
  - verify は、*file_name* 内の値の検証を実行します。このサブコマンドは、不正な値を指定してサーバを設定するのを防ぐので、設定ファイルを編集した場合に役立ちます。

- restore は、最新の設定値を使用して *file_name* を作成します。静的パラメータの新しい値を設定した後でこのサブコマンドを使用すると、現在の実行値ではなく、設定した値がファイルに書き込まれます。このコマンドは、設定ファイルのコピーがすべて消失してしまった場合に新しいコピーを生成するときに便利です。*file_name* のディレクトリを指定しない場合は、ファイルは Adaptive Server が起動されたディレクトリに書き込まれます。

- *file_name* には、*subcommand* とともに使用する設定ファイルを指定します。ファイル名の一部としてディレクトリを指定しない場合は、Adaptive Server の起動ディレクトリが使用されます。

例

例 1 *srv.config* ファイル内の値の検証を実行し、検証をパスしたパラメータをサーバに読み込みます。検証をパスしない値に対しては現在の実行値が使用されます。

```
sp_configure "configuration file", 0, "read", "srv.config"
```

例 2 *my_server.config* ファイルを作成し、サーバが使用している現在の設定値をこのファイルに書き込みます。

```
sp_configure "configuration file", 0, "write", "my_server.config"
```

設定ファイルの編集

設定ファイルは ASCII ファイルであるので、テキスト・エディタを使用して編集し、ASCII フォーマットで保存することができます。それぞれのパラメータの構文は、次のとおりです。

parameter_name={*value* | DEFAULT}

構文の説明は、次のとおりです。

- *parameter_name* は、指定するパラメータの名前です。
- *value* は、指定した *parameter_name* を設定する数値です。
- “DEFAULT” は、*parameter_name* にデフォルト値を使用する場合に指定します。

例

例 1 次の例では、インデックスのページ分割または縮小中にデッドロックが発生した場合に、トランザクションがロックの取得を 1 回だけリトライするように指定します。

```
deadlock retries = 1
```

例 2 次の例では、パラメータ cpu accounting flush interval にはデフォルト値を使用することを指定します。

```
cpu accounting flush interval=DEFAULT
```

設定ファイルを編集する場合は、verify オプションを使用してファイルを検証するか、read オプションを使用してファイルを読み込むか、またはその設定ファイルを指定して Adaptive Server を再起動するまでは、編集内容の検証は行われません。

すべての設定ファイルが消失するか破損した場合は、restore サブコマンドを使用することによって、稼働中のサーバから設定ファイルを作り直して、新しいファイル名を指定できます。新しいファイル内のパラメータは、サーバが現在実行している値に設定されます。

設定ファイルに対するパーミッション

設定ファイルは暗号化されていない ASCII テキスト・ファイルです。デフォルトでは、ファイルの所有者に対しては読み取りと書き込みのパーミッションが、他のすべてのユーザに対しては読み取りパーミッションが設定されて作成されます。オペレーティング・システム・レベルで設定ファイルを作成する場合は、作成者がファイルの所有者になります。write または restore パラメータを使用して Adaptive Server から設定ファイルを作成する場合は、ファイルの所有者は Adaptive Server を起動したユーザになります。通常、これは “sybase” というユーザです。設定ファイルへのアクセスを制限するには、オペレーティング・システムのファイル・パーミッション・コマンドを使用して、読み込み、書き込み、実行の許可を設定してください。

注意 パーミッションは、作成した設定ファイルごとに設定する必要があります。

設定ファイルのバックアップ

master データベースのバックアップを実行しても、設定ファイルは自動的にはバックアップされません。設定ファイルはオペレーティング・システム・ファイルであるため、他のオペレーティング・システム・ファイルをバックアップするのと同じ方法でバックアップしてください。

現在使用されている設定ファイル名の確認

sp_configure の出力では、表示領域の制限により設定ファイル名はトランケートされます。設定ファイル名を完全に表示するには、次の構文を使用します。

```
select s1.value2
from syscurconfigs s1, sysconfigures s2
where s1.config = s2.config
and s2.name = "configuration file"
```

設定ファイルを指定した Adaptive Server の起動

デフォルトでは、Adaptive Server の起動時に、起動ディレクトリ内にある設定ファイル *server_name.cfg* が読み込まれます。このファイルが存在しない場合は、新しいファイルが作成され、すべての値にデフォルト値が使用されます。

Adaptive Server の起動時に設定ファイルを指定することができます。詳細については、『ASE ユーティリティ・ガイド』を参照してください。

指定した設定ファイルが存在しない場合、Adaptive Server はエラー・メッセージを表示し、起動しません。

コマンドが正常に処理されると、ファイル *server_name.bak* が作成されます。このファイルには、指定した設定ファイルから読み込んだ値で sysconfigures が更新される前に、sysconfigures 内に保管されていた設定値が含まれます。このファイルは以降の起動のたびに上書きされます。

設定ファイルのエラー

設定ファイルにエラーがある場合は、Adaptive Server が起動しないか、デフォルト値が使用されるかのいずれかとなります。

次の場合はデフォルト値が使用されます。

- 無効な値がある場合。たとえば、数値を必要としているパラメータに対して、設定ファイルでは文字列が指定されている場合は、デフォルト値が使用されます。
- 値が最小許容値より小さい場合。

パラメータの階層

設定パラメータは、対象となる Adaptive Server の動作領域に従ってグループ分けされています。このグループ分けによって、Adaptive Server の特定の領域のパフォーマンスを改善するために調整を必要とする、すべてのパラメータを簡単に識別できます。

パラメータはそれぞれ 1 つのプライマリ・グループに属していますが、多くのパラメータはセカンダリ・グループにも属しています。たとえば、number of remote connections は、ネットワーク通信グループにプライマリとして属するとともに、メモリ使用グループにもセカンダリとして属します。これは、パラメータの中には Adaptive Server の複数の動作領域に関連するものがあることを表しています。sp_configure を実行すると、パラメータは所属するすべてのグループに表示されます。

表 5-2 は、設定パラメータのグループを示します。

表 *5-2:* 設定グループ

パラメータ・グループ	Adaptive Server の設定対象
Backup/Recovery	データのバックアップとリカバリ
Cache manager	データ・キャッシュとプロシージャ・キャッシュ

パラメータ・グループ	Adaptive Server の設定対象
Component Integration Services administration	コンポーネント統合サービス
DTM administration	DTM (分散トランザクション管理) 機能
Diagnostics	診断の原則
Disk I/O	ディスク I/O
Error log	Windows イベント・ログに記録される Adaptive Server のエラー・メッセージとイベント・ログ
Extended stored procedures	ESP (拡張ストアド・プロシージャ) の動作
General information	システムの基本的な管理
Java services	Adaptive Server 内の Java のメモリ データベースにおける Java の詳細については、『Adaptive Server Enterprise における Java』を参照してください。 JDBC へのメソッド呼び出しを使用する場合には、ユーザが使用できる実行スタックのサイズを大きくしなければならない場合があります。「stack size」(236 ページ) を参照してください。
Languages	言語、ソート順、文字セット
Lock manager	ロック
Memory use	メモリの消費
Metadata caches	頻繁に使用されるシステム・カタログ情報用のメタデータ・キャッシュ・サイズの設定。「メタデータ・キャッシュ」は、データベース、インデックス、オブジェクトに関する情報の追跡に使用する、予約済みのメモリ領域です。オープンしているデータベース、インデックス、オブジェクトの数が多いほど、メタデータ・キャッシュ・サイズは大きくなります。メモリ使用に関して使用されるメタデータ・キャッシュの詳細については、『システム管理ガイド 第 2 巻』の「第 3 章 メモリの設定」を参照してください。
Monitoring	モニタリング情報の収集。デフォルトでは、Adaptive Server はモニタリング情報を収集しません。 『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』の「第 2 章 モニタリング・テーブルの詳細」を参照してください。
Network communication	Adaptive Server とリモート・サーバ間、Adaptive Server とクライアント・プログラム間の通信
O/S resources	オペレーティング・システム・リソースの使用
Physical memory	マシンの物理メモリ・リソース
Processors	SMP 環境のプロセッサ
Query Tuning	クエリの最適化
RepAgent thread administration	Replication Server を介した複写
SQL Server administration	Adaptive Server の全般的な管理
Security related	セキュリティ関連機能
Unicode	Unicode 関連機能
User environment	ユーザ環境

すべてのグループとそのグループに属するパラメータ、およびパラメータの現在の値を表示するための構文は次のとおりです。

```
sp_configure
```

注意 sp_configure が返すパラメータの数は、設定した表示レベルの値によって変わります。「パラメータ階層のユーザ定義サブセット：表示レベル」(71 ページ) を参照してください。

特定のグループとそのグループに属するパラメータを表示するための構文は次のとおりです。

```
sp_configure "group_name"
```

たとえば、ディスク I/O (Disk I/O) グループを表示するには次のように入力します。

```
sp_configure "Disk I/O"

Group: disk I/O
Parameter Name           Default Memory Used Config Value Run Value
unit         type
--------------           ------- ----------- ------------ ---------
------       -------------
allow sql server async i/o     1           0            1         1
switch       static
diable disk mirroring          1           0            1         1
switch       static
disk i/o structures          256           0          256       256
number       dynamic
number of devices             10           0           10        10
number       dynamic
number of large I/O buffers    6       12352            6         6
number       dynamic
page utilization percent      95           0           95        95
percent      dynamic
```

注意 サーバで大文字と小文字を区別しないソート順が使用されている場合に、パラメータを指定しないで sp_configure を実行すると、グループ分けは表示されず、すべての設定パラメータとグループのアルファベット順リストが返されます。

パラメータ階層のユーザ定義サブセット：表示レベル

Adaptive Server の使用形態によっては、一部のパラメータを他のパラメータよりも頻繁に調整することが必要です。パラメータのサブセットを操作する方が簡単な場合があります。

デフォルトの表示レベルは、「包括」です。設定した表示レベルは、以降のセッションにも適用されます。ただし、いつでも再設定できます。

- basic (基本) − 最も基本的な設定パラメータだけを表示し、一般的なサーバ・チューニングに適しています。
- intermediate (中間) − 基本レベルで表示されるパラメータに加えて、やや複雑なパラメータも表示されます。
- comprehensive (包括) − 最も複雑なものまでを含むすべてのパラメータが表示されます。このレベルは、かなり詳細なサーバ調整を行う場合に適しています。

現在の表示レベルを表示するための構文は、次のとおりです。

```
sp_displaylevel
```

表示レベルを設定するための構文は次のとおりです。

```
sp_displaylevel user_name[, basic | intermediate | comprehensive]
```

user_name は、実行するユーザの Adaptive Server ログイン名です。

sp_configure 出力への表示レベルの影響

表示レベルが「基本レベル」か「中間レベル」に設定されている場合に sp_configure を実行すると、返されるパラメータは、「包括レベル」の場合に返されるパラメータのサブセットだけです。たとえば、表示レベルが「中間レベル」に設定されているときに、言語グループ内のパラメータを表示するには、次のように入力します。

```
sp_configure "Languages"
```

出力は次のようになります。

```
sp_configure
Group: 言語

Parameter Name         Default Memory Used Config Value Run Value Unit Type
---------------        ------- ----------- ------------ --------- ---- ----
default character set    1          0            1          1     id   static
default language id      0          0            0          0     id   dyna
. . .
```

ここで表示されるのは、言語グループ内のパラメータのサブセットだけです。一部の言語パラメータは、表示レベルが「包括レベル」のときだけ表示されます。

sp_configure と *sp_sysmon* によるパフォーマンス・チューニング

sp_sysmon は、Adaptive Server のパフォーマンスをモニタし、Adaptive Server システムの動作を表す統計情報を生成します。『パフォーマンス&チューニング・シリーズ：sp_sysmon による Adaptive Server の監視』を参照してください。

sp_configure を使用する前と使用後に sp_sysmon を実行して、設定パラメータを調整できます。出力からパフォーマンス・チューニングの基礎情報が得られ、設定変更の結果を監視できます。

クラスタード環境における設定パラメータの使用

クラスタ・エディションでは、クラスタ全体とインスタンス固有の設定の両方がサポートされています。クラスタ全体設定パラメータは、クラスタ内のインスタンスすべてに適用されます。ローカル設定パラメータは、指定されたインスタンスだけに適用されます。

- ローカル設定は、クラスタ全体設定を無効にする。
- インスタンス固有の設定が適用されていない場合、クラスタ全体設定が適用される。
- 一部のパラメータ (default character set id など) は、特定のインスタンスに適用できない。それらのパラメータはクラスタ全体にだけ適用できる。

クラスタ設定ファイルには、インスタンス固有の設定ブロックがあります。インスタンス固有ブロックのパラメータ設定は、クラスタ全体設定を無効にします。次に例を示します。

```
max online engines = DEFAULT

[Instance:ase1]
max online engines = 5
[Instance:ase2]
max online engines = 3
```

『Cluster ユーザーズ・ガイド』を参照してください。

sp_configure 出力

次の出力例は、表示レベルを「包括レベル」に設定し、パラメータを指定しないで sp_configure を実行した場合に出力される情報を示しています。出力される値は、プラットフォームや既に変更している値によって異なります。

```
sp_configure
Group: Configuration Options

Group: バックアップとリカバリ

Parameter Name         Default Memory Used Config Value Run Value Unit Type
--------------         ------- ----------- ------------ --------- ---- ----
allow remote access          1           0            1         1    switch dyn
print recovery info          0           0            0         0    switch dyn
recovery interval in m       5           0            5         5    minutes dyn
...
```

注意　表示レベルを「包括レベル」に設定している場合は、すべての設定グループとパラメータが出力されます。

各カラムの内容は、次のとおりです。

- “Default” カラムにはデフォルト値が表示されます。パラメータを明示的に再設定しなければ、そのパラメータにはデフォルト値が使用されます。
- “Memory Used” カラムには現在の値のパラメータが使用しているメモリの量が、キロバイト単位で表示されます。関連するパラメータどうしが、同じメモリ・プールのメモリを使用することがあります。たとえば、stack size と stack guard size が使用するメモリは、number of user connections が使用するメモリの一部として含まれています。これらの各パラメータが使用するメモリを別々に加算すると、総計は実際に使用されているメモリ量よりも多くなります。他のパラメータとメモリを共有しているパラメータには、シャープ記号 (#) が付きます。
- “Config Value” カラムには、設定パラメータの最新の値が表示されます。sp_configure を実行して動的パラメータを変更すると、次のようになります。
  - 設定値と実行値が更新されます。
  - 設定ファイルが更新されます。
  - 変更はすぐに有効になる。

  静的パラメータを変更すると、次のようになります。
  - 設定値が更新されます。
  - 設定ファイルが更新されます。
  - 変更内容は、Adaptive Server を再起動しないと有効にならない。

- “Run Value” カラムには Adaptive Server が現在使用している値が表示されます。この値は、動的パラメータの場合は値を変更したときに変更されます。静的パラメータの場合は、Adaptive Server を再起動したときに変更されます。
- “Unit” カラムには設定パラメータの値の単位が表示されます。Adaptive Server の情報の表示に使用される単位は次のとおりです。

単位名	説明
number	項目数。
clock ticks	クロック・チック数。
microseconds	マイクロ秒数。
milliseconds	ミリ秒数。
seconds	秒数。
minutes	分数。
hours	時数。
bytes	バイト数。
days	日数。
kilobytes	キロバイト数。
megabytes	メガバイト数。
memory pages (2K)	2K のメモリ・ページ数。
virtual pages (2K)	2K の仮想ページ数。
logical pages	論理ページ数。この値は、サーバが使用している論理ページ・サイズ (2K、4K、8K、または 16K) によって異なる。2、4 ～ 65535、またはシステム制限値
percent	パーセントとして設定されたパラメータの値。
ratio	比率として設定されたパラメータの値。
switch	TRUE (オン) または FALSE (オフ) として設定されているパラメータの値。
id	調査している設定パラメータの ID。
name	パラメータの実行値または設定値に割り当てられた文字列名。たとえば、sp_configure “lock scheme” の出力の “ Run Value” カラムまたは “Config Value” カラムに “binary” という文字列が表示される。
row	ロー数。

- “Type” カラムには、設定オプションが静的と動的のどちらであるかが表示されます。静的パラメータへの変更を有効にするには、Adaptive Server を再起動する必要があります。動的パラメータへの変更はただちに反映されるため、Adaptive Server を再起動する必要はありません。

Named Cache 設定パラメータ

Named Cache 設定パラメータ・グループは、名前付きキャッシュの詳細を提示します。

- cache size － キャッシュのサイズ。デフォルトでは、Adaptive Server は 8MB キャッシュを作成します。sp_cacheconfig を使用してこのパラメータを動的に変更するか、サーバ設定ファイルの値を変更すると、サーバの再起動後に変更が反映されます。
- cache status － キャッシュのステータス。取り得る値は、`default data cache`、`log only`、`mixed`、`in-memory storage` のいずれかです。キャッシュのデフォルトのステータスは、`default data cache` でなければならず、これは変更できません。名前付きキャッシュの cache status は、`log only` か `mixed` であり、イン・メモリ・データベースでは `in-memory storage` (イン・メモリ・データベースの cache status は変更できません) です。

  他のインスタンスが別の cache status を使用している間は、ローカル・キャッシュ上でクラスタード環境における cache status を `log only` から動的に変更できません。
- cache replacement － キャッシュ置換方式を記述する。名前付きキャッシュとデフォルトのデータ・キャッシュの場合、置換方式は `strict LRU` (ストリクト LRU) か `relaxed LRU` (リラックス LRU) です。sp_cacheconfig を使用してこのパラメータを動的に変更するか、サーバ設定ファイルの値を変更すると、サーバの再起動後に変更が反映されます。イン・メモリ・データベースの場合、キャッシュ置換方式は `none` (なし) でなければなりません。イン・メモリ・データベースでは、バッファもページ置換も使用しないからです。
- local cache partition number － キャッシュ・パーティション数。1 つの名前付きキャッシュを複数のキャッシュ・パーティションに分割できます。使用できる値は 0、2、4、8、16、32、64、128 です。キャッシュ・パーティション数は動的に変更できません。変更を反映するには、Adaptive Server を再起動する必要があります。

sysconfigures テーブルと syscurconfigs テーブル

sp_configure によって表示されるレポートは、主に master..sysconfigures システム・テーブルと master..syscurconfigs システム・テーブルから生成され、sysattributes、sysdevices などのシステム・テーブルの追加情報が含まれます。

sysconfigures テーブルの value カラムには、sp_configure または設定ファイルによって設定した最終値が記録されます。syscurconfigs の value カラムには、現在使用されている値が格納されます。動的パラメータの場合、2 つの値は一致します。静的パラメータは、サーバを再起動するまで有効にならないため、Adaptive Server の起動後に値を変更した場合は、2 つの値は異なります。デフォルト値を使用している場合も値が異なることがあります。この場合、sysconfigures には 0 が格納され、syscurconfigs には Adaptive Server が計算して使用している値が格納されます。

sp_configure は、sp_configure によってレポートされた値を表示するために、sysconfigures と syscurconfigs でジョインを実行します。

syscurconfigs と sysconfigures へのクエリ (例)

sysconfigures と syscurconfigs に対してクエリを実行し、独自に編成した情報を取得できます。たとえば、引数を指定しないで sp_configure を実行すると、設定パラメータに使用されているメモリはリストされますが、最小値と最大値はリストされません。次のクエリを使用して、最大値、最小値、デフォルト値以外に、現在のメモリ使用量もすべて記載した一覧を取得します。

```
select b.name, memory_used, minimum_value,
maximum_value, defvalue
from master.dbo.sysconfigures b,
master.dbo.syscurconfigs c
where b.config *= c.config and parent != 19
and b.config > 100
```

設定パラメータ

多くの場合、設定パラメータの最大許容値は、sp_configure の制限値ではなく使用できるメモリによって制限されます。

注意 プラットフォームおよび Adaptive Server のバージョンに応じた設定可能な最大値については、使用しているプラットフォームの『インストール・ガイド』の「Adaptive Server の仕様」を参照してください。

設定パラメータのアルファベット順リスト

それぞれの設定パラメータの要約と詳細について、以降の項で説明します。

abstract plan cache

要約	
デフォルト値	0 (オフ)
値の範囲	0 (オフ)、1 (オン)
ステータス	動的
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	クエリ・チューニング

abstract plan cache パラメータは、抽象プラン・ハッシュ・キーのキャッシングを有効にします。詳細については、『パフォーマンス&チューニング・シリーズ：クエリ処理と抽象プラン』の「第 12 章 抽象プランの作成と使用」を参照してください。プランのキャッシングを有効にするには、abstract plan load を有効にする必要があります。

abstract plan dump

要約	
デフォルト値	0 (オフ)
値の範囲	0 (オフ)、1 (オン)
ステータス	動的
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	クエリ・チューニング

abstract plan dump パラメータは、ap_stdout 抽象プラン・グループへの抽象プランの保存を有効にします。詳細については、『パフォーマンス&チューニング・シリーズ：クエリ処理と抽象プラン』の「第 12 章 抽象プランの作成と使用」を参照してください。

abstract plan load

要約	
デフォルト値	0 (オフ)
値の範囲	0 (オフ)、1 (オン)
ステータス	動的
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	クエリ・チューニング

abstract plan load パラメータは、ap_stdin 抽象プラン・グループの抽象プランとクエリとの関連付けを有効にします。詳細については、『パフォーマンス&チューニング・シリーズ：クエリ処理と抽象プラン』の「第 12 章 抽象プランの作成と使用」を参照してください。

abstract plan replace

要約	
デフォルト値	0 (オフ)
値の範囲	0 (オフ)、1 (オン)
ステータス	動的
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	クエリ・チューニング

abstract plan replace パラメータは、ap_stdout 抽象プラン・グループの抽象プランのプラン置換を有効にします。詳細については、『パフォーマンス&チューニング・シリーズ：クエリ処理と抽象プラン』の「第 12 章 抽象プランの作成と使用」を参照してください。置換モードを有効にするには、abstract plan load を有効にする必要があります。

additional network memory

要約	
デフォルト値	0
値の範囲	0 ～ 2147483647
ステータス	動的
表示レベル	中間
必要な役割	システム管理者
設定グループ	メモリ使用、ネットワーク通信、物理メモリ

additional network memory は、ネットワーク・パケットのサイズがデフォルトのパケット・サイズより大きい場合に使用される追加メモリの最大サイズを設定します。入力した値は、2K の倍数になるように切り捨てられます。デフォルト値は、大きなパケット用の追加領域は割り付けられていないことを意味します。

ログイン時にデフォルトよりも大きなパケット・サイズの要求があった場合、Adaptive Server は、要求に応えることが可能なメモリを使用できるかどうかを調べます。メモリが不足している場合は、使用可能なメモリの中から最大サイズのブロックが検出され、最大メモリ・ブロックを下回る適切なサイズ (default network packet size の倍数) のメモリを使用できるかどうかが検証されます。使用できない場合は、要求から default network packet size 分のメモリが差し引かれ、その値に相当するメモリが使用できるかどうかが確認されます。メモリが使用可能になるか、引き算の結果が default network packet size と同じになるまで、この引き算処理が最大 10 回繰り返されます。引き算処理が 10 回に達すると、default network packet size の値がパケット・サイズとして使用されます。

max network packet size の値を増やす場合、additional network memory の値も増やす必要があります。割り付けられたネットワーク・メモリはすべて、デフォルト・サイズでユーザ用に予約されているからです。デフォルトのパケット・サイズであれば、すべてのユーザ接続が Adaptive Server に確実にログインできます。

max network packet size を増やしたが、additional network memory を増やさなかった場合、デフォルト・サイズより大きいネットワーク・パケット・サイズを要求するクライアントは、必ずしも要求したパケット・サイズでログインできるとは限りません。

additional network memory を増やすと、大量のデータを転送するアプリケーションの場合はパフォーマンスを改善できることがあります。アプリケーションが大きなパケット・サイズを使用する場合に additional network memory の値を決めるには、次の手順に従います。

1　大きなパケット・サイズを要求する同時ユーザの数と、そのアプリケーションが要求するサイズを見積もります。

2　それぞれの接続は 3 つのバッファを必要とするので、この合計を 3 倍します。

3　32 ビット・サーバでは 2 パーセント、64 ビット・サーバでは 4 パーセントの値をオーバヘッドとして加算します。

4　値が 2048 の倍数になるように切り上げます。

次の例は、複数のアプリケーションが大きいパケット・サイズを必要とする場合の見積もりを示します。

アプリケーション	パケット・サイズ　オーバヘッド
bcp	8192
Client-Library	8192
Client-Library	4096
Client-Library	4096
合計	24576

アプリケーション	パケット・サイズ	オーバヘッド
3 バッファ／ユーザを乗算	* 3=73728	
2% のオーバヘッドを計算		* .02=1474
オーバヘッドを加算	+ 1474	
追加のネットワーク・メモリ	75202	
2,048 の倍数になるように切り上げる	75776	

additional network memory を 75,776 バイトに設定します。

allocate max shared memory

要約	
デフォルト値	0 (オフ)
値の範囲	0 (オフ)、1 (オン)
ステータス	動的
表示レベル	基本
必要な役割	システム管理者
設定グループ	メモリ使用、物理メモリ

allocate max shared memory は、Adaptive Server の起動時に max memory で指定されたメモリをすべて割り付けるか、設定パラメータに必要な量のメモリだけを割り付けるかを指定します。

allocate max shared memory を 0 に設定すると、Adaptive Server が使用する共有メモリの量は現在の設定に必要な量だけとなり、起動時に割り付けられるのは設定パラメータが必要とする量だけ、つまり max memory の値よりは少なくなります。

allocate max shared memory を 1 に設定すると、max memory に指定された量のメモリがすべて Adaptive Server の起動時に割り付けられます。allocate max shared memory を 1 に設定し、max memory を増やすと、Adaptive Server はメモリをすぐに割り付けようとします。メモリの割り付けに失敗した場合、メッセージがエラー・ログに書き込まれます。エラー・ログをチェックして、エラーが何も発生していないことを確認してください。

メモリ割り付けに成功した場合、Adaptive Server はメモリ設定がどのように変更されても必要なメモリを常に確保します。また、サーバがメモリ追加の調整を行っている間にパフォーマンスが低下することはありません。ただし、メモリの増加量を正確に予測できない場合、max memory を大きな値に設定すると、物理メモリを浪費するおそれがあります。

allow backward scans

要約	
デフォルト値	1 (オン)
有効な値	0 (オフ)、1 (オン)
ステータス	動的
表示レベル	中間
必要な役割	システム管理者
設定グループ	クエリ・チューニング

allow backward scans パラメータは、order by...desc コマンドが含まれる select クエリをオプティマイザが実行する方法を制御します。

- この値が 1 に設定されているときは、オプティマイザはページ・チェーンをインデックスの降順に検索してインデックスまたはテーブル・ローにアクセスできます。
- この値が 0 に設定されているときは、オプティマイザはインデックス・ページ・ポインタを昇順にたどりながらローを選択してワークテーブルに入れ、このワークテーブルを降順でソートします。

最初の方法、つまり後方スキャンを使用すれば、結果をカラム値の降順で並べる必要がある場合にテーブルへのアクセスを高速化できます。ただし、アプリケーションによっては、後方スキャンによるデッドロックが発生することがあります。特に、同じインデックスを使用して前方スキャンを行う delete、または update クエリがある場合は、デッドロックが増加しているかどうかを調べてください。また、インデックス内のページ分割によるデッドロックが発生することもあります。

print deadlock information パラメータを使用して、デッドロックについてのメッセージをエラー・ログに送信します。「print deadlock information」(212 ページ) を参照してください。または、システム・プロシージャ sp_sysmon を使用してデッドロックがあるかどうかを確認します。詳細については、『パフォーマンス＆チューニング・シリーズ : ロックと同時実行制御』を参照してください。

allow nested triggers

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

allow nested triggers パラメータは、ネストされたトリガの使用を制御します。値を 1 に設定すると、トリガによるデータ変更で別のトリガを起動することができます。ネストされたトリガを使用できないようにするには、allow nested triggers を 0 に設定します。set の self_recursion オプションは、トリガによる変更で、そのトリガを再び起動できるようにするかどうかを制御します。

allow procedure grouping

要約	
デフォルト値	1 (オン)
値の範囲	0 (オフ)、1 (オン)
ステータス	動的
表示レベル	包括
必要な役割	システム・セキュリティ担当者
設定グループ	セキュリティ関連

allow procedure grouping パラメータは、同じ名前のストアド・プロシージャを 1 つの drop procedure 文で削除できるようにグループ化する機能を制御します。

allow remote access

要約	
デフォルト値	1 (オン)
有効な値	0 (オフ)、1 (オン)
ステータス	動的
表示レベル	中間
必要な役割	システム・セキュリティ担当者
設定グループ	バックアップとリカバリ、ネットワーク通信

allow remote access は、リモートの Adaptive Server からのログインを制御します。デフォルト値は 1 で、このとき Adaptive Server は Backup Server と通信できます。

注意 この値を 0 に設定すると、サーバ間のリモート・プロシージャ・コール (RPC) は使用できなくなります。Adaptive Server は RPC を使用して Backup Server と通信するので、このパラメータを 0 に設定すると、データベースのバックアップが実行できなくなります。

Backup Server 以外のリモート・サーバが RPC を実行できるようにするには、この他にもシステム管理作業が必要であるので、このオプションの設定を 1 のままにしておいてもセキュリティ上の危険はありません。

allow resource limits

要約	
デフォルト値	0 (オフ)
有効な値	0 (オフ)、1 (オン)
ステータス	静的
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	メモリ使用、SQL Server 管理

allow resource limits パラメータは、リソース制限の使用を制御します。値を 1 に設定すると、サーバは時間範囲、リソース制限、内部サーバ・アラーム用に内部メモリを割り付けます。また、サーバは、ユーザ・セッションに対して適用可能な範囲と制限を内部的に割り当てます。showplan と statistics io の出力には、オプティマイザによるクエリの見積もりコストが表示されます。リソース制限をすべて無効にするには、allow resource limits を 0 に設定してください。

allow sendmsg

要約	
デフォルト値	0 (オフ)
有効な値	0 (オフ)、1 (オン)
ステータス	動的
表示レベル	包括
必要な役割	システム・セキュリティ担当者
設定グループ	ネットワーク通信

allow sendmsg は、Adaptive Server から UDP (User Datagram Protocol) ポートへのメッセージ送信を有効または無効にします。allow sendmsg を 1 に設定すると、すべてのユーザが sp_sendmsg または syb_sendmsg を使用してメッセージを送信できます。Adaptive Server が使用するポート番号の設定については、「syb_sendmsg port number」(242 ページ) を参照してください。

注意　UDP ポートへのメッセージ送信は Windows ではサポートされていません。

allow sql server async i/o

要約	
デフォルト値	1 (オン)
有効な値	0 (オフ)、1 (オン)
ステータス	静的
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	ディスク I/O

allow sql server async i/o は、Adaptive Server が非同期ディスク I/O で実行できるようにします。オペレーティング・システム・レベルで非同期 I/O を使用可能にするための情報については、それぞれのオペレーティング・システムのマニュアルを参照してください。

ディスク I/O は同期式よりも非同期式の方が必ず高速に実行されます。これは、Adaptive Server が非同期 I/O を発行するときに、応答を待たずに次の I/O を発行できるためです。

allow updates to system tables

要約	
デフォルト値	0 (オフ)
有効な値	0 (オフ)、1 (オン)
ステータス	動的
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	SQL Server 管理

allow updates to system tables パラメータは、システム管理者の役割を持つユーザがシステム・テーブルを変更したり、システム・テーブルを変更できるストアド・プロシージャを作成したりできるようにします。allow updates to system tables が有効になっている場合、データベース管理者は自分が所有するすべてのテーブル内のシステム・テーブルを更新できます。

システム・テーブルには次のものがあります。

- master データベース内の Sybase が提供するすべてのテーブル

- 名前が “sys” で始まり、sysobjects テーブルでの ID 値が 100 以下である、ユーザ・データベース内のすべてのテーブル。

警告！ システム・テーブルに対する変更が正しくない場合は、データベースが破壊されてデータが消失することがあります。システム・テーブルを変更する場合は、データベースを破壊させる可能性があるエラーから保護するために、常に begin transaction を使用してください。変更を終了したら、すぐに allow updates to system tables を無効にしてください。

allow updates to system tables パラメータが “on” に設定されている間に作成されたストアド・プロシージャとトリガは、このパラメータを “off” にした後でも、システム・テーブルを更新できます。allow updates to system tables を “on” に設定すると、その間はユーザがシステム・テーブルを変更したり、後でシステム・テーブルの変更に使用できるストアド・プロシージャを作成したりできるので、「脆弱な時間帯」を作り出すことになります。

システム・テーブルは非常に重要なので、十分に制御された状況下以外ではこのパラメータを “on” にしないことをおすすめします。システム・テーブルの直接更新が可能である間は Adaptive Server に他のユーザが一切アクセスできないようにするために、Adaptive Server をシングルユーザ・モードで再起動します。詳細については、『ユーティリティ・ガイド』の startserver と dataserver の説明を参照してください。

注意 サーバ全体の設定オプションallow updates to system tables は、allow updates to system tables のストアド・プロシージャ設定よりも優先されます。サーバ・レベルで allow updates to system tables を有効にしていない場合、システム・カタログを変更できるかどうかは、個々のストアド・プロシージャ設定によって決まります。

average cap size

要約	
デフォルト値	200
値の範囲	100 ～ 10000
ステータス	静的
表示レベル	
必要な役割	
設定グループ	診断

今後のために予約済み。

audit queue size

要約	
デフォルト値	100
値の範囲	1 ～ 65535
ステータス	動的
表示レベル	中間
必要な役割	システム・セキュリティ担当者
設定グループ	メモリ使用、セキュリティ関連

メモリ内の監査キューは、ユーザ処理によって生成された監査レコードの処理と監査証跡への書き込みが可能になるまで、そのレコードを保持します。システム・セキュリティ担当者は、audit queue size を使用して監査キューのサイズを変更できます。キューのサイズを設定するときは、パフォーマンスとリスクの間のトレードオフがあります。キューが大きすぎると、レコードが長い間キューの中にとどまることがあります。レコードがキューの中にある間は、システムに障害が発生した場合に消失する危険があります。しかし、キューが小さすぎるとすぐに空きがなくなり、システム全体のパフォーマンスが低下します。監査キューの空きがなくなると、監査レコードを生成するユーザの処理はスリープします。

監査キューの大きさを決定するためのガイドラインには、次のようなものがあります。また、実行する監査の量も考慮する必要があります。

- 1 つの監査レコードに必要なメモリは 424 バイトですが、データ・ページに書き込まれるときのレコードは 22 バイトまで小さくできます。
- システム障害で消失する監査レコードの最大数は、監査キューのサイズ (レコード単位) に 20 を加えた値です。監査キューから取り除かれたレコードは、ディスク上の現在の監査テーブルに書き込まれるまでバッファ・ページに残ります。ページは 20 レコードごとにディスクにフラッシュされます (監査プロセスがビジー状態になることが少なければ、20 レコード未満でもフラッシュされます)。
- システム監査テーブル内の extrainfo フィールドと名前用のフィールドは可変長であるため、名前情報をすべて記録する監査レコードは一般に大きくなります。

1 ページに収まる監査レコードの数は、4 から 80 程度までの間で変化します。デフォルトの監査キュー・サイズ 100 に対するメモリ必要量は約 42K です。

auditing

要約	
デフォルト値	0 (オフ)
値の範囲	0 (オフ)、1 (オン)
ステータス	動的
表示レベル	中間
必要な役割	システム・セキュリティ担当者
設定グループ	セキュリティ関連

auditing パラメータは、Adaptive Server の監査を有効または無効にします。

automatic cluster takeover

要約	
デフォルト値	1
有効な値	1 (有効)、0 (無効)
ステータス	動的
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	共有ディスク・クラスタ

automatic cluster takeover を 1 に設定すると、起動中のインスタンスが突然の全体的クラスタ障害から自動的にリカバリできます。automatic cluster takeover を 0 に設定すると、--cluster_takeover パラメータを含めないかぎり、クラスタは突然のクラスタ・フェールオーバからリカバリできないことがあります。

クラスタ・エディションは、クォーラム・ハートビートとクラスタ・テイクオーバ・アルゴリズムを使用して、クラスタ・テイクオーバをいつ実行すべきかを判断します。このアルゴリズムでは、起動中のインスタンスがクラスタにジョインできないのは、クラスタがクラッシュしたため (この場合にはテイクオーバが適切) なのか、それともネットワーク接続がないため (この場合にはテイクオーバは不適切) なのかを区別することができます。

automatic cluster takeover が無効にされる (0 に設定) と、クラスタ・エディションはアルゴリズムの結果をアドバイス・メッセージとしてエラー・ログに書き込んでから、終了します。

auotomatic cluster takeover が有効にされる (1 に設定) と、クラスタ・エディションはクラスタ・コーディネータとして起動し、データベースをリカバリします。これは、I/O フェンシングが有効に設定された環境で安全な操作であることが保証されています。

I/O フェンシングがない環境では、アルゴリズムに不具合があるとデータ破損が発生する可能性があるので、この設定パラメータを 0 に設定してこのアルゴリズムを無効に設定することをおすすめします。ただし、I/O フェンシングがない環境はデータ破損の危険性があるので、自動クラスタ・テイクオーバ機能を無効にしても、この危険性がすべて低減されるわけではありません。

builtin date strings

要約	
デフォルト値	0
値の範囲	0 ～ 1
ステータス	動的
表示レベル	
必要な役割	
設定グループ	クエリ・チューニング

日付順の値ではなく文字列が引数として指定された場合、サーバは示された精度にかかわらず、その文字列を datetime 値として解釈します。デフォルトの動作は、設定パラメータ builtin date strings または設定オプション builtin_date_strings を設定することで変更できます。これらのオプションを設定すると、サーバは日付順の組み込みに提示された文字列を bigdatetime として解釈します。

caps per ccb

要約	
デフォルト値	50
値の範囲	5 ～ 50
ステータス	静的
表示レベル	
必要な役割	
設定グループ	診断

今後のために予約済み。

check password for digit

要約	
デフォルト値	0 (オフ)
値の範囲	1 (オン)、0 (オフ)
ステータス	動的
表示レベル	10
必要な役割	システム・セキュリティ担当者
設定グループ	セキュリティ関連

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、パスワードに文字か数字が 1 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。

CIPC large message pool size

要約	
デフォルト値	512
有効な値	512 ～ 2147483647
ステータス	静的
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	共有ディスク・クラスタ

CIPC large message pool size は、起動時に CIPC によって割り付けられる大きいメッセージのバッファの数を指定します。

CIPC regular message pool size

要約	
デフォルト値	8192
有効な値	2048 ～ 2147483647
ステータス	静的
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	共有ディスク・クラスタ

CIPC regular message pool size は、起動時に CIPC によって割り付けられる通常サイズのメッセージのバッファの数を指定します。

cis bulk insert array size

要約	
デフォルト値	50
値の範囲	0 ～ 2147483647
ステータス	動的
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	コンポーネント統合サービス

Adaptive Server から別の Adaptive Server へのデータのバルク転送を実行するとき、CIS はローを内部的にバッファし、Open Client バルク・ライブラリに対してバッファ内のローを 1 つのブロックとして転送するように要求します。配列のサイズは cis bulk insert array size で設定します。

cis bulk insert batch size

要約	
デフォルト値	0
値の範囲	0 ～ 2147483647
ステータス	動的
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	コンポーネント統合サービス

cis bulk insert batch size は、select into を使用してソース・テーブルからターゲット・テーブルに 1 つのバッチとしてバルク・コピーされるローの数を決定します。

cis bulk insert batch size を 0 のままにしておくと、すべてのローが 1 つのバッチとしてコピーされます。0 以外の場合、このパラメータに指定した数のローがターゲット・テーブルにコピーされた後、サーバがターゲット・サーバにバルク・コミットを発行することにより、バッチがコミットされます。

クライアントが生成する通常のバルク・コピー操作 (bcp ユーティリティで生成されるような) を受け取ったときは、クライアントがバルク・バッチのサイズを制御するものと見なされ、サーバはこの設定パラメータの値を無視します。

cis connect timeout

要約	
デフォルト値	0
値の範囲	0 ～ 32767
ステータス	動的
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	コンポーネント統合サービス

cis connect timeout は、Client-Library 接続が正常に完了するまでの待ち時間を秒単位で定義します。

cis cursor rows

要約	
デフォルト値	50
値の範囲	1 ～ 2147483647
ステータス	動的
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	コンポーネント統合サービス

cis cursor rows は、cursor open 操作と cursor fetch 操作でのカーソル・ロー・カウントを指定します。この値を増やすと、1 つの操作でより多くのローがフェッチされるようになります。これによって処理速度は速くなりますが、メモリがより多く必要になります。

cis idle connection timeout

要約	
デフォルト値	0
値の範囲	0 ～ 2147483647
ステータス	動的
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	コンポーネント統合サービス

cis idle connection timeout は、指定された秒数よりも長い時間未使用になっているリモート・サーバへの CIS 接続がないか確認するように Adaptive Server を設定します。Adaptive Server は未使用の接続を削除し、それらのリソースを再割り付けします。

指定する数値は秒単位ですが、ハウスキーピング・タスクは 1 分間に多くても 1 回しかウェイクアップしないので、アイドル接続は設定された値よりもかなり長くアイドル状態になる場合があります。トランザクションが接続でアクティブな場合、アイドル接続は削除されず、ユーザが接続にアクセスするためにコマンドを実行すると自動的に接続が再確立されます。

cis packet size

要約	
デフォルト値	512
値の範囲	512 ～ 32768
ステータス	動的
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	コンポーネント統合サービス

cis packet size は、サーバとリモート・サーバの間で接続開始時に交換される TDS (Tabular Data Stream™) パケットのサイズを指定します。

ほとんどのシステムではデフォルトのパケット・サイズは 512 バイトであり、これはほとんどのアプリケーションに十分な値です。ただし、特に text、unitext、image データまたはバルク・データが関係する場合は、パケット・サイズをこれより大きくするとクエリのパフォーマンスが大幅に向上することがあります。

デフォルトよりも大きいパケット・サイズを指定する場合は、ターゲット・サーバで可変長のパケット・サイズを処理できるように設定します。次のとおりにします。

- additional netmem
- maximum network packet size

cis rpc handling

要約	
デフォルト値	0 (オフ)、クラスタ・エディションでは 1
有効な値	0 (オフ)、1 (オン)
ステータス	動的
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	コンポーネント統合サービス

cis rpc handling は、リモート・プロシージャ・コール (RPC) のデフォルトの処理方法を指定します。cis rpc handling を 0 に設定すると、Adaptive Server のサイト・ハンドラがデフォルトの RPC 処理メカニズムとして設定されます。このパラメータを 1 に設定すると、RPC 処理にはコンポーネント統合サービスのアクセス・メソッドが使用されます。『コンポーネント統合サービス・ユーザーズ・ガイド』の cis rpc handling の設定の説明を参照してください。

cluster heartbeat interval

要約	
デフォルト値	10
有効な値	1 ～ 127
ステータス	動的
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	共有ディスク・クラスタ

cluster heartbeat interval は、クラスタ・インスタンスがハートビート・ステータスの送信とチェックに使用する間隔を制御します。

cluster heartbeat interval に低い値を指定すると、障害検出時間が短縮されますが、一時的な問題 (CPU のオーバーロードなど) が原因で誤検出の危険性が高まります。cluster heartbeat interval を高い値に調整すると、誤検出の危険性は低下しますが、障害の検出に要する時間が長くなります。

cluster heartbeat retries

要約	
デフォルト値	1
有効な値	1 ～ 127
ステータス	動的
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	共有ディスク・クラスタ

cluster heartbeat retries は、インスタンスが障害モードに入るまでに、失敗したクラスタ・ハートビートを再試行する回数を制御します。

cluster heartbeat retries を低い値に調整すると、障害検出時間が短縮されますが、一時的な問題 (CPU のオーバーロードなど) が原因で誤検出の危険性が高まります。cluster heartbeat retries を高い値に調整すると、誤検出の危険性は低下しますが、障害の検出に要する時間が長くなります。

cluster vote timeout

要約	
デフォルト値	60
有効な値	1 ～ 127
ステータス	動的
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	共有ディスク・クラスタ

cluster vote timeout は、あるインスタンスが投票期間中に他のインスタンスが投票するのを待つ最長時間を制御します。インスタンスが待つのは、稼働中だと思われるインスタンスだけです。

cluster vote timeout を低い値に調整すると、フェールオーバ時間が短縮されますが、稼働中のインスタンスが新しいクラスタ・ビューから除外されてしまう危険性が高まります。cluster vote timeout を高い値に調整すると、稼働中のインスタンスが新しいクラスタ・ビューから除外されてしまう危険性は低下しますが、フェールオーバ時間が長くなる可能性があります。

compression memory size

要約	
デフォルト値	0
値の範囲	0 ～ 2147483647
ステータス	動的
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	物理メモリ

圧縮ダンプをアーカイブ・データベースにロードする間に使用されます。compression memory size は、Adaptive Server が圧縮ダンプを圧縮解除するために使用するメモリ・プールのサイズ (2KB ページ単位) を決定します。compression memory size が 0 に設定された場合、プールは作成されず、圧縮ダンプをロードできません。

『システム管理ガイド 第 2 巻』の「第 14 章 アーカイブ・データベースへのアクセス」の「圧縮メモリ・プールの作成」を参照してください。

configuration file

要約	
デフォルト値	0 (オフ)
値の範囲	0、verify、read、write、restore のいずれか
ステータス	動的
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	一般情報

configuration file は、現在使用中の設定ファイルのロケーションを指定します。構成ファイルの詳細については、「設定ファイルを指定して sp_configure を使用する方法」(64 ページ) を参照してください。

sp_configure で出力される “Run Value” カラムには 10 文字しか表示されません。そのため、設定ファイルのパスと名前を完全な形では表示できない場合があります。

cost of a logical io

要約	
デフォルト値	2
値の範囲	0 ～ 254
ステータス	動的
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	クエリ・チューニング

cost of a logical io は、1 つの論理 I/O のコストを指定します。

cost of a physical io

要約	
デフォルト値	25
値の範囲	0 ～ 254
ステータス	動的
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	クエリ・チューニング

cost of a phsyical io は、1 つの物理 I/O のコストを指定します。

cost of a cpu unit

要約	
デフォルト値	1000
値の範囲	1 ～ 65534
ステータス	動的
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	クエリ・チューニング

cost of a cpu unit は、1 つの CPU 処理のコストを指定します。

オプティマイザの逐次プランのコストを求める公式は、次のとおりです。

```
Cost = PIO X estimated_pio + LIO X estimated_lio + 100 X estimated_cpu / CPU
```

デフォルト値は次のとおりです。

- *estimated_pio* = 25
- *estimated_lio* = 2
- *estimated_cpu* = 1000

Adaptive Server にメモリが十分にある場合、テーブルはすべてメモリ内に存在します。cost of a physical io の値は 0 が適切です。

CPU が十分に高速であるため cost of a cpu unit の値が問題にならない場合には、この公式を使用して CPU のコストを求めます。これは、2 LIO と 25 PIO (いずれもデフォルト値) を組み合わせたものです。

```
CPU X 100/configuration_value
```

configuration_value のデフォルト値は 1000 です。

cost of a cpu unit の値を増やすにつれて、CPU がコストに及ぼす影響は小さくなっていきます。

cpu accounting flush interval

要約	
デフォルト値	200
値の範囲	1 ～ 2147483647
ステータス	動的
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	SQL Server 管理

cpu accounting flush interval パラメータは、Adaptive Server が各ユーザの CPU 使用統計を sysprocesses から syslogins にフラッシュするまでの待機時間を、マシンのクロック・チック (Adaptive Server のクロック・チックではない) 単位で指定します。これは、チャージバック・アカウンティングで使用される手順です。

ユーザが Adaptive Server にログインすると、その時点以降のそのユーザ・プロセスによる CPU 使用量が sysprocesses に蓄積されます。ユーザが Adaptive Server からログオフするか、cpu accounting flush interval の時間が過ぎると、蓄積された CPU 使用統計情報は sysprocesses から syslogins にフラッシュされます。この統計情報は、合計がクリアされるまでは、syslogins に引き続き蓄積されます。syslogins から現在の合計を表示するには、sp_reportstats を使用します。

cpu accounting flush interval に設定する値は、目的とするレポートのタイプによって異なります。レポート処理の頻度が低ければ、syslogins 内のデータを頻繁に更新することはそれほど重要ではありません。

ただし、プロセスによる CPU の使用量を調べるために、アドホック・クエリを使用して syslogins の totcpu カラムからの選択を定期的に実行する場合は、cpu accounting flush interval の設定値を小さくします。このようにすれば、選択を実行するときに syslogins 内のデータが最新のものである可能性が高くなります。

cpu accounting flush interval の設定値を小さくすると、プロセスがデッドロックのビクティムの候補であるとロック・マネージャが誤って判断することがあります。ロック・マネージャは、デッドロックを検出すると、競合するプロセスのそれぞれによって蓄積された CPU 使用時間をチェックします。この値が小さい方のプロセスがデッドロック・ビクティムとして選択され、ロック・マネージャによって終了させられます。また、cpu accounting flush interval の設定値が小さいと、プロセスの CPU 使用情報を保管するタスク・ハンドラが初期化される頻度が上がるので、プロセスが実際に使用した CPU 時間よりも蓄積された時間が少ないように認識されることがあります。このため、実際にはプロセスが蓄積した CPU 使用時間が競合するプロセスよりも多いにもかかわらず、ロック・マネージャがそのプロセスをデッドロック・ビクティムとして選択することがあります。

CPU 使用時間のレポートがまったく必要ない場合は、cpu accounting flush interval を最大値に設定してください。これにより、syslogins が更新される回数と、そのページをディスクに書き込まなければならない回数が減ります。

cpu grace time

要約	
デフォルト値	500
値の範囲	0 ～ 2147483647
ステータス	動的
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	SQL Server 管理

cpu grace time パラメータは、time slice パラメータとともに使用します。この時間が経過するまではユーザ・プロセスは CPU を解放することなく実行できますが、この値を超えると、Adaptive Server はそのプロセスの制御を横取りしてタイムスライス・エラーで終了させます。cpu grace time の単位は、sql server clock tick length で定義されているタイム・チックです。「sql server clock tick length」(232 ページ) を参照してください。

プロセスの実行時間が cpu grace time を超過すると、Adaptive Server は内部キューからプロセスを取り除くことによってそのプロセスに「影響を及ぼし」ます。プロセスは強制終了されますが、Adaptive Server は影響を受けません。こうして、プロセスの暴走によって CPU が独占されるのを防ぎます。ユーザ・プロセスの中断が発生した場合は、cpu grace time の値を増やすことによって一時的にこの問題を避けることができます。ただし、問題が本当に暴走しているプロセスによるものではなく、現在の cpu grace time の範囲内で完了できないプロセスのためであることを確認する必要があります。

cpu grace time を一時的に増やすことは対処方法ではありますが、他の問題を引き起こす可能性があるため、永久的な解決策ではありません。これについては、「time slice」(246 ページ) を参照してください。また、タスク・スケジューリングの詳細については、『パフォーマンス&チューニング・シリーズ：基本』の「第 4 章 エンジンと CPU の使用方法」を参照してください。

current audit table

要約	
デフォルト値	1
値の範囲	0 ～ 8
ステータス	動的
表示レベル	中間
必要な役割	システム・セキュリティ担当者
設定グループ	セキュリティ関連

current audit table パラメータは、Adaptive Server が監査ローを書き込むテーブルを設定します。システム・セキュリティ担当者は、次の構文を使用して現在の監査テーブルを変更できます。

```
sp_configure "current audit table", n
  [, "with truncate"]
```

ここで n は、次に示すように現在の新しい監査テーブルを決定する整数値です。

- 1 は sysaudits_01、2 は sysaudits_02 を意味し、最大 8 まで設定できます。
- 0 は、次のテーブルを現在の監査テーブルとするように Adaptive Server に指示します。たとえば、インストール環境に 3 つの監査テーブル sysaudits_01、sysaudits_02、sysaudits_03 がある場合、現在の監査テーブルは次のように設定されます。

- 現在の監査テーブルが sysaudits_01 の場合は 2
- 現在の監査テーブルが sysaudits_02 の場合は 3
- 現在の監査テーブルが sysaudits_03 の場合は 1

“with truncate” は、新しいテーブルが空でない場合に、そのテーブルをトランケートすることを指定します。このオプションが指定されていないときに、テーブルが空でなければ、sp_configure コマンドは失敗します。

注意　Adaptive Server が現在の監査テーブルをトランケートしたときに、データがアーカイブ済みでなければ、そのテーブルの監査レコードは失われます。監査データがアーカイブされていることを確認してから、with truncate オプションを使用してください。

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。

deadlock checking period

要約	
デフォルト値	500
値の範囲	0 ～ 2147483
ステータス	動的
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	ロック・マネージャ

deadlock checking period パラメータは、ロックの解放を待っているプロセスに対して Adaptive Server がデッドロックのチェックを開始するまでの最小時間を、ミリ秒単位で指定します。このデッドロックのチェックは、デッドロックがまったく発生しないか、少しだけ発生するアプリケーションにとっては時間のかかるオーバヘッドであり、ロックを待つ必要があるロック要求の割合が増えるに従って、オーバヘッドが増加します。

deadlock checking period を 0 以外の値 (*n*) に設定すると、プロセスの待ち時間が *n* ミリ秒以上となったときにデッドロックのチェックが開始します。たとえば、次のように入力することで、プロセスがロックを待つ時間が 700 ミリ秒に達してからデッドロックがチェックされるように設定できます。

```
sp_configure "deadlock checking period", 700
```

deadlock checking period パラメータを 0 に設定すると、それぞれのプロセスがロック待ち状態となると同時にデッドロックのチェックが開始します。クロック・チックのミリ秒数より小さい値はすべて 0 と見なされます。「sql server clock tick length」(232 ページ) を参照してください。

deadlock checking period を大きな値に設定すると、デッドロックが検出されるまでの時間が長くかかります。ただし、設定された時間が経過する前にほとんどのロック要求が受け入れられるので、それらのロック要求に対するデッドロックのチェックのためのオーバヘッドは回避されます。アプリケーションでのデッドロックの頻度が低い場合は、deadlock checking period を高い値に設定します。それ以外の場合は、デフォルト値の 500 ミリ秒で十分です。

使用しているシステムにおけるデッドロックの頻度と deadlock checking period パラメータの最適な設定を判断するには、sp_sysmon を使用してください。『パフォーマンス＆チューニング・シリーズ：sp_sysmon による Adaptive Server の監視』を参照してください。

deadlock pipe active

要約	
デフォルト値	0
値の範囲	0 ～ 1
ステータス	動的
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	メモリ使用、モニタリング

deadlock pipe active は、Adaptive Server でデッドロック・メッセージを収集するかどうかを制御します。deadlock pipe active と deadlock pipe max messages を両方とも有効にすると、Adaptive Server は各デッドロックのテキストを収集します。収集されたデッドロック・メッセージは、monDeadLock を使用して取得できます。

deadlock pipe max messages

要約	
デフォルト値	0
値の範囲	0 ～ 2147483647
ステータス	動的
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	モニタリング

deadlock pipe max messages は、Adaptive Server が格納するデッドロック・メッセージ数をエンジンごとに決定します。monSQLText テーブル内のメッセージ数の合計は、sql text pipe max messages に実行中のエンジン数を掛け合わせた値になります。

deadlock retries

要約	
デフォルト値	5
値の範囲	0 ～ 2147483647
ステータス	動的
表示レベル	中間
必要な役割	システム管理者
設定グループ	ロック・マネージャ、SQL Server 管理

deadlock retries パラメータは、インデックスの**ページ分割**または縮小中にデッドロックが発生した場合にトランザクションがロックの取得を試行できる回数を指定します。

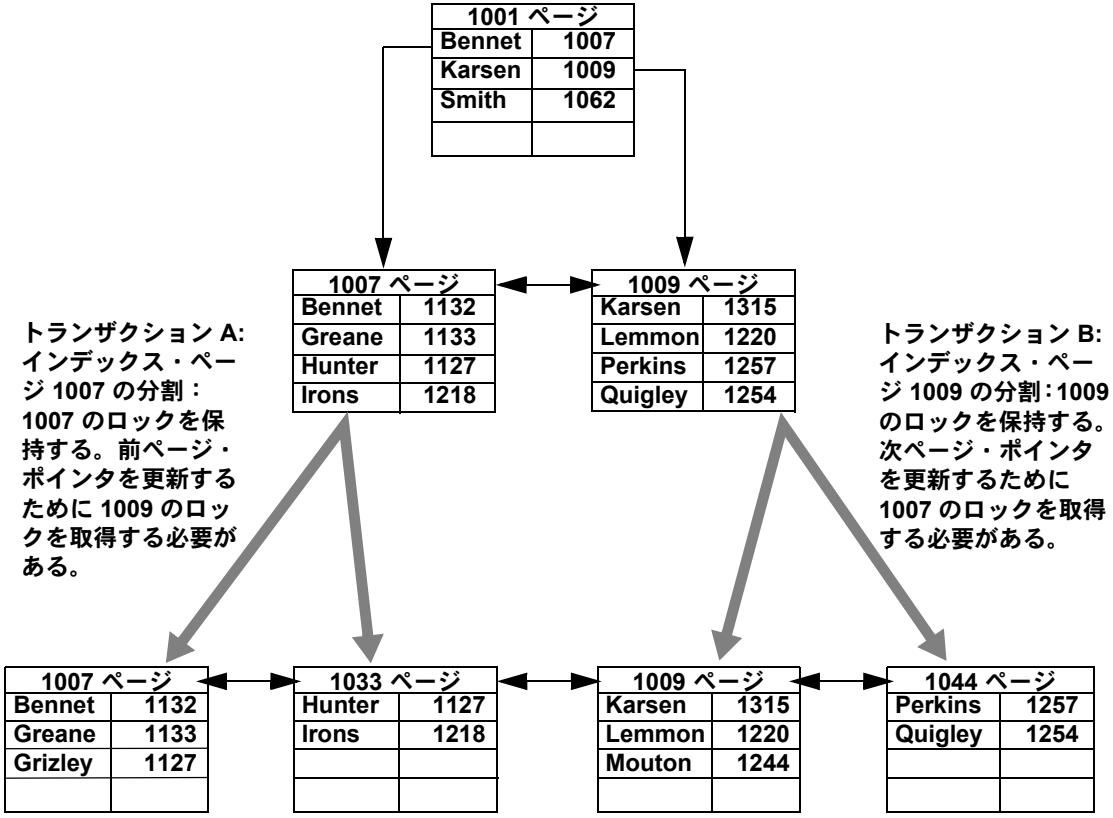
たとえば、図 5-1 は次の状況を示しています。

- トランザクション A はページ 1007 をロックします。また、ページ分割用のページ・ポインタを更新するために、ページ 1009 のロックを取得する必要があります。
- トランザクション B もインデックス・ローを挿入するのでページ分割が発生し、ページ 1009 をロックします。また、ページ 1007 のロックを取得する必要があります。

この状況では、Adaptive Server はデッドロックのビクティムとなるプロセスをすぐに選択するのではなく、いずれかのトランザクションのインデックス・ロックを解放させます。ほとんどの場合、これによって他方のトランザクションは完了し、ロックを解放できます。

ロックの試行を放棄するトランザクションでは、インデックスはルート・ページから再スキャンされ、ページ分割オペレーションは deadlock retries で指定されている回数だけ再試行されます。

図 *5-1:* クラスタード・インデックス内でページ分割中のデッドロック



sp_sysmon は、デッドロックとデッドロック・リトライをレポートします。詳細については、『パフォーマンス&チューニング・シリーズ：ロックと同時実行制御』を参照してください。

default character set id

要約	
デフォルト値	1
値の範囲	0 ～ 255
ステータス	静的
表示レベル	中間
必要な役割	システム管理者
設定グループ	言語

default character set id パラメータは、サーバによって使用されるデフォルトの文字セットの番号を指定します。デフォルトはインストール時に設定されますが、Sybase インストール・ユーティリティを使用して後で変更できます。「第 9 章 文字セット、ソート順、言語の設定」を参照してください。

default database size

要約	
デフォルト値	3MB
値の範囲	2[a] ～ 10000 a. 最小値はサーバの論理ページ・サイズによって決まる
ステータス	動的
表示レベル	中間
必要な役割	システム管理者
設定グループ	SQL Server 管理

default database size パラメータは、create database 文にサイズ・パラメータの指定をせずに発行した場合に、新しいユーザ・データベースに割り付けられるデフォルト・サイズをメガバイト単位で設定します。create database 文で指定されたデータベース・サイズは、この設定パラメータによって設定する値よりも優先します。

新規データベースのほとんどが論理ページ 1 ページ分よりも多くの領域を必要とする場合は、デフォルト・データベース・サイズの値を大きくする必要があります。

注意　create database コマンドはmodel データベースをコピーして新しいユーザ・データベースを作成するので、model データベースを変更した場合は、default database size も増やす必要があります。

default exp_row_size percent

要約	
デフォルト値	5
値の範囲	0 ～ 100
ステータス	動的
表示レベル	中間
必要な役割	システム管理者
設定グループ	SQL Server 管理

default exp_row_size percent パラメータは、データオンリーロック・テーブルでの拡張更新用に領域を予約します。この目的は、ローの転送を減らすことです。「拡張更新」とは、ローの長さが増えるような、データ・ローへの更新のことです。null 値を持つことができるデータ・ローや可変長カラムのあるデータ・ローへの更新は、拡張更新となる可能性があります。データオンリーロック・テーブルで拡張更新が発生したとき、データ・ローのサイズが増えてそのページに収まらなくなると、ローの転送が必要になることがあります。

デフォルト値を使用する場合は、使用可能なデータ・ページ・サイズの 5 パーセントが拡張更新用に確保されます。データオンリーロック・テーブルのページではデータの記憶領域用に 2,002 バイトが使用可能なので、拡張用には 100 バイトが残されます。この値が適用されるのは、可変長カラムのあるテーブルのページに対してだけです。

default exp_row_size percent を 0 に設定すると、すべてのページが最後まで使用され、拡張更新用の領域は残されません。

exp_row_size が create table に明示的に指定されていない場合や sp_chgattribute で設定されている場合、default exp_row_size percent パラメータは可変長カラムのあるデータオンリーロック・テーブルに適用されます。create table で指定された値は、設定パラメータの設定値よりも優先されます。詳細については、『パフォーマンス&チューニング・シリーズ：ロックと同時実行制御』を参照してください。

default fill factor percent

要約	
デフォルト値	0
値の範囲	0 ～ 100
ステータス	動的
表示レベル	中間
必要な役割	システム管理者
設定グループ	SQL Server 管理

default fill factor percent パラメータは、既存のデータに対する新しいインデックスを作成するときに create index 文でフィルファクタ (fillfactor) が指定されなかった場合に、各インデックス・ページにどの程度までデータを格納するかを決定します。fillfactor の値は、インデックスを作成するときにだけ使用されます。データは変更されるので、ページが特定の満杯率で維持されることはありません。

default fill factor percent パラメータは、次のものに影響を与えます。

- データが使用する記憶領域の量 － Adaptive Server は、クラスタード・インデックスを作成するときにデータを再分配します。
- パフォーマンス － ページの分割は Adaptive Server のリソースを消費します。

この値よりも create index コマンドの指定が優先するので、default fill factor percent パラメータを変更する必要はほとんどありません。『リファレンス・マニュアル：コマンド』の「create index」を参照してください。

default language id

要約	
デフォルト値	0
値の範囲	0 ～ 32767
ステータス	動的
表示レベル	中間
必要な役割	システム管理者
設定グループ	言語

default language id は、サーバで使用できる言語の中から別の言語をユーザが選択していない場合に、システム・メッセージの表示に使用される言語の番号です。us_english の ID は常に NULL です。言語を追加すると、そのときにユニークな番号がその言語に割り当てられます。

default network packet size

要約	
デフォルト値	2048
値の範囲	512 ～ 65024
ステータス	静的
表示レベル	中間
必要な役割	システム管理者
設定グループ	メモリ使用、ネットワーク通信、ユーザ環境

default network packet size は、すべての Adaptive Server ユーザに対するデフォルトのパケット・サイズを設定します。default network packet size に設定できる値は、512 バイトの倍数だけです。それ以外の値を指定した場合は、512 バイトの整数倍になるように切り捨てられます。

デフォルトのパケット・サイズでログインするすべてのユーザ用のメモリは、total logical memory で設定される Adaptive Server のメモリ・プールから割り付けられます。このメモリは、Adaptive Server の起動時にネットワーク・パケット用に割り付けられます。

それぞれの Adaptive Server ユーザ接続は次のバッファを使用します。

- 1 つの読み込みバッファ
- 1 つのメッセージ用バッファ
- 1 つの書き込みバッファ

これらのバッファはそれぞれ、default network packet size で設定されるバイト数を必要とします。ネットワーク・パケット用に割り付けられるメモリの総量は次のとおりです。

```
(number of user connections + number of worker processes) * 3 * default network packet size
```

たとえば、default network packet size の設定値が 1024 バイトで、50 のユーザ接続と 20 のワーカー・プロセスがある場合は、必要なネットワーク・メモリの量は次のとおりです。

(50 + 20) * 3 * 1024 = 215040 バイト

default network packet size の値を大きくした場合は、max network packet size もそれと同じサイズ以上に増やす必要があります。max network packet size の値が default network packet size の値より大きい場合は、additional network memory の値を増やしてください。「additional network memory」(78 ページ) を参照してください。

default network packet size パラメータの変更が、ネットワーク I/O 管理とタスク切り替えにどのように影響するかを確認するには、sp_sysmon を使用してください。たとえば、default network packet size を増やしてから、sp_sysmon の出力をチェックすることにより、bcp で大きいバッチを処理するときにこの設定がどのように影響するかを確認します。『パフォーマンス&チューニング・シリーズ：sp_sysmon による Adaptive Server の監視』を参照してください。

ログイン時のパケット・サイズ増加の要求

bcp や isql などのほとんどのクライアント・プログラムでは、デフォルトのパケット・サイズは 512 バイトに設定されています。デフォルトのパケット・サイズを変更するには、クライアントの接続時にそれより大きいパケット・サイズを要求する必要があります。Adaptive Server のクライアント・プログラムで -A フラグを使用すると、デフォルトよりも大きなパケット・サイズを要求できます。次に例を示します。

```
isql -A2048
```

default sortorder id

要約	
デフォルト値	50
値の範囲	0 ～ 255
ステータス	静的
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	言語

default sortorder id は、サーバにデフォルトとして現在インストールされているソート順の番号です。デフォルト・ソート順を変更するには、「第 9 章 文字セット、ソート順、言語の設定」を参照してください。

default unicode sortorder

要約	
デフォルト値	バイナリ
値の範囲	現在は使用されていない
ステータス	静的
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	Unicode

default unicode sortorder は、サーバにインストールされている Unicode のデフォルトのソート順をユニークに定義する文字列パラメータです。Unicode のデフォルト・ソート順を変更するには、「第 9 章 文字セット、ソート順、言語の設定」を参照してください。

default XML sortorder

要約	
デフォルト値	バイナリ
値の範囲	(現在は使用されていない)
ステータス	静的
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	Unicode

default XML sortorder は、XML エンジンによって使用されるソート順を定義する文字列パラメータです。『Adaptive Server Enterprise における XML サービス』の「第 6 章 XML における国際化のサポート」を参照してください。

deferred name resolution

要約	
デフォルト値	0 (無効)
値の範囲	0 ～ 1
ステータス	動的
必要な役割	システム管理者
設定グループ	クエリ・チューニング

deferred name resolution がアクティブ (1) の場合、遅延名前解決がサーバ接続すべてにグローバルに適用されます。サーバで作成するプロシージャはすべて、遅延名前解決を使用して作成されます。

したがって、ストアド・プロシージャは、ストアド・プロシージャ内で参照されるオブジェクトを解決することなく作成され、オブジェクト解決処理は実行時まで延期されます。『Transact-SQL ユーザーズ・ガイド』の「第 17 章 ストアド・プロシージャの使用」を参照してください。

disable character set conversions

要約	
デフォルト値	0 (有効)
有効な値	0 (有効)、1 (無効)
ステータス	静的
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	言語

disable character set conversions を 1 に変更すると、クライアントと Adaptive Server との間でやり取りされるデータの文字セット変換がオフになります。たとえば、あるクライアントが Latin-1 (iso_1) を使用していて、Adaptive Server がデフォルトの文字セットとして Roman-8 (roman8) を使用している場合は、クライアントのデータは Adaptive Server にロードされるときに Roman-8 に変換されます。Latin-1 を使用しているクライアントに送信されるデータは再変換されますが、Adaptive Server と同じ文字セットを使用しているクライアントの場合には変換されません。

disable character set conversions を設定することにより、変換を行わないことを指定できます。たとえば、すべてのクライアントが同じ文字セットを使用していて、Adaptive Server ではその文字セットですべてのデータが保存されるようにするには、disable character set conversions を 1 に設定すれば変換は行われません。

disable disk mirroring

要約	
デフォルト値	1
有効な値	0 (オフ)、1 (オン)
ステータス	静的
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	ディスク I/O

disable disk mirroring は、Adaptive Server のディスク・ミラーリングを有効または無効にします。この設定パラメータはグローバル変数であるため、設定パラメータを 1 に設定して Adaptive Server を再起動した後は、ディスク・ミラーリングは一切実行されません。disable disk mirroring を 0 に設定すると、ディスク・ミラーリングが有効になります。

注意　Adaptive Server でフェールオーバを使用できるように設定されている場合は、ディスク・ミラーリングを無効にする必要があります。

disk i/o structures

要約	
デフォルト値	256
値の範囲	0 ～ 2147483647
ステータス	動的
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	ディスク I/O、メモリ使用

disk i/o structures は、Adaptive Server が起動時に割り付けるディスク I/O 制御ブロック数の初期値を指定します。

Adaptive Server がユーザ・プロセスの I/O 要求を開始するには、そのプロセス用のディスク I/O 制御ブロックが必要です。ディスク I/O 制御ブロック用のメモリは、Adaptive Server の起動時に事前に割り付けられています。ディスク I/O 構造体が不足することがないようにするには、disk i/o structures をオペレーティング・システムで許容される最大の値に設定します。同時ディスク I/O 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。

ディスク I/O 構造体をさらに割り付ける必要があるかどうかを判断するには、sp_sysmon を使用してください。『パフォーマンス＆チューニング・シリーズ：sp_sysmon による Adaptive Server の監視』を参照してください。max async i/os per server 設定パラメータは、disk i/o structures と同じ値に設定できます。「max async i/os per server」(151 ページ) を参照してください。

DMA object pool size

要約	
デフォルト値	4096
有効な値	2048 ～ 2147483647
ステータス	静的
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	共有ディスク・クラスタ

DMA object pool size は、起動時に CIPC によって割り付けられるダイレクト・メモリ・アクセス (DMA: Direct Memory Access) オブジェクトの数を指定します。

dtm detach timeout period

要約	
デフォルト値	0 (分)
有効な値	0 ～ 2147483647 (分)
ステータス	動的
表示レベル	10
必要な役割	システム管理者
設定グループ	DTM 管理

dtm detach timeout period は、分散トランザクション分岐を分離した状態で保持できる時間を分単位で設定します。X/Open XA 環境では、トランザクションは制御スレッドから分離する場合があります。分離は、一般には別の制御スレッドに付加するために行います。dtm detach timeout period で指定された時間は、トランザクションを分離した状態に保持できます。この時間が過ぎると、Adaptive Server は分離されたトランザクションをロールバックします。

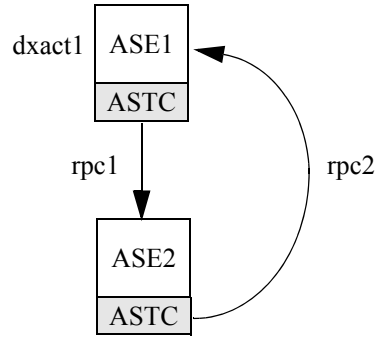
dtm lock timeout period

要約	
デフォルト値	300 (秒)
有効な値	1 ～ 2147483647 (秒)
ステータス	動的
表示レベル	10
必要な役割	システム管理者
設定グループ	DTM 管理

dtm lock timeout period は、ロック・リソースが使用可能になるまで分散トランザクション分岐が待機する最大時間を秒単位で設定します。この時間が経過すると、Adaptive Server はトランザクションがデッドロック状態にあると見なし、デッドロックを引き起こしたトランザクション分岐をロールバックします。これにより、最終的に分散トランザクション全体がロールバックされます。

リモート・サーバにトランザクションが送信された後で、このリモート・サーバから発信サーバにトランザクションが返信されると、分散トランザクション自体がデッドロックすることがあります。この状況を図 5-2 に示します。分散トランザクション “dxact1” の作業が、“rpc1” を経由して Adaptive Server 2 に送信されます。このとき、Adaptive Server 2 は “rpc2” を経由してトランザクションをコーディネーティング・サーバに返信します。“rpc2” と “dxact1” は、同じ gtrid を共有しますが、分岐修飾子が異なるので、同じトランザクション・リソースを共有することはできません。“rpc2” が、“dxact1” によって保持されているロックを待機している場合は、デッドロック状態が発生します。“rpc2” が、“dxact1” によって保持されているロックを待機している場合は、デッドロック状態が発生します。

図 *5-2:* 分散トランザクション・デッドロック



Adaptive Server は、サーバ間のデッドロックを検出できません。その代わりに、dtm lock timeout period に依存します。図 5-2 では、dtm lock timeout period の期間を過ぎると、“rpc2” に対して作成されたトランザクションがアボートされます。その結果 Adaptive Server 2 は作業での失敗をレポートし、最終的に “dxact1” もアボートされます。

dtm lock timeout period の値は、分散トランザクションだけに適用されます。ローカル・トランザクションでは、サーバワイドの lock wait period パラメータで指定されるロック・タイムアウト時間を使用できます。

注意　システム・テーブル上のデッドロックの検出には dtm lock timeout period は使用されません。

---

dump on conditions

要約	
デフォルト値	0 (オフ)
値の範囲	0 (オフ)、1 (オン)
ステータス	動的
表示レベル	中間
必要な役割	システム管理者
設定グループ	グループ診断

dump on conditions パラメータは、maximum dump conditions パラメータで指定された状態になったときに、ダンプ・データを共有メモリ内に生成するかどうかを決定します。

注意 dump on conditions パラメータは、Sybase 製品の保守契約を結んでいるサポート・センタだけが使用します。このパラメータは、Sybase 製品の保守契約を結んでいるサポート・センタから指示がないかぎり、変更しないでください。

dynamic allocation on demand

要約	
デフォルト値	1 (オン)
値の範囲	0 (オフ)、1 (オン)
ステータス	動的
表示レベル	基本
必要な役割	システム管理者
設定グループ	メモリ使用、物理メモリ

dynamic allocation on demand は、動的メモリ設定パラメータの変更に応じていつメモリを割り付けるかを指定します。

dynamic allocation on demand を 1 に設定すると、メモリは必要になったときにのみ割り付けられます。たとえば、number of user connections の設定を 100 から 200 に変更した場合は、各ユーザ用のメモリはそのユーザがサーバに接続するまでは追加されません。Adaptive Server は、変更後の最大ユーザ接続数に達するまでは、メモリの追加を続けます。

dynamic allocation on demand を 0 に設定すると、動的設定パラメータの変更によって必要となるメモリがすべて即時に割り付けられます。したがって、ユーザ接続の最大数を 100 から 200 に変更した場合には、追加された 100 のユーザ接続に必要なメモリがただちに割り付けられます。

enable backupserver HA

要約	
デフォルト値	1
有効な値	1 (有効)、0 (無効)
ステータス	動的
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	共有ディスク・クラスタ

enable backupserver HA を 1 に設定すると、クラスタの高可用性 Backup Server が起動します。enable backupserver HA を 0 に設定すると、クラスタ上の高可用性 Backup Server が無効になります。

enable cis

要約	
デフォルト値	1 (オン)
有効な値	0 (オフ)、1 (オン)
ステータス	静的
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	コンポーネント統合サービス

enable cis は、コンポーネント統合サービスを有効または無効にします。

enable DTM

要約	
デフォルト値	0 (オフ)
有効な値	0 (オフ)、1 (オン)
ステータス	静的
表示レベル	10
必要な役割	システム管理者
設定グループ	DTM 管理、SQL Server 管理

enable DTM は、Adaptive Server 分散トランザクション管理 (DTM: Distributed Transaction Management) 機能を有効または無効にします。DTM 機能が有効の場合は、Adaptive Server を X/Open XA システムや MSDTC システムのリソース・マネージャとして使用できます。サーバを再起動すると、このパラメータが有効になります。X/Open XA 環境での Adaptive Server の使用方法については、『XA インタフェース統合ガイド for CICS、Encina、TUXEDO』を参照してください。MSDTC 環境でのトランザクションおよび Adaptive Server ネイティブのトランザクション・コーディネーション・サービスについては、『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。

enable encrypted columns

要約	
デフォルト値	0 (オフ)
値の範囲	0 (オフ)、1 (オン)
ステータス	動的
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	セキュリティ関連

enable encrypted columns は、暗号化カラムを有効にします。

ASE_ENCRYPTION ライセンスを購入してサーバへのインストールと登録を完了しないかぎり、enable encrypted columns を設定できません。ライセンスなしに設定しようとすると、メッセージ 10834 が表示されます。

```
Configuration parameter 'enable encrypted columns' cannot be
enabled without license 'ASE_ENCRYPTION'
```

注意 暗号化カラムを使用すると、論理メモリ使用量が 8198 キロバイト増加します。

enable enterprise java beans

要約	
デフォルト値	0 (オフ)
値の範囲	0 (オフ)、1 (オン)
ステータス	静的
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	Java サービス

enable enterprise java beans は、Adaptive Server データベースで EJB サーバを使用できるかどうかを指定します。Adaptive Server での EJB サーバの使用を有効にしなければ、EJB サーバは使用できません。

enable file access

要約	
デフォルト値	1 (オン)
有効な値	0 (オフ)、1 (オン)
ステータス	静的
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	コンポーネント統合サービス

enable file access は、プロキシ・テーブルを介した外部ファイル・システムへのアクセスを有効にします。ASE_XFS のライセンスが必要です。

enable full-text search

要約	
デフォルト値	1
有効な値	0 (オフ)、1 (オン)
ステータス	静的
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	コンポーネント統合サービス

enable full-text search は、拡張型全文検索サービスを有効にします。ASE_EFTS のライセンスが必要です。

enable HA

要約	
デフォルト値	0 (オフ)
値の範囲	0 ～ 2
ステータス	静的
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	SQL Server 管理

enable HA を 1 に設定すると、Adaptive Server をアクティブ／アクティブな高可用性サブシステムのコンパニオン・サーバとして設定できます。enable HA を 2 に設定すると、Adaptive Server をアクティブ／パッシブな高可用性サブシステムのコンパニオン・サーバとして設定できます。

Adaptive Server は、Sybase のフェールオーバ機能を使用して高可用性サブシステムと連動します。enable HA を 1 に設定してから、*installhasvss* スクリプト (Windows では *insthasv*) を実行してください。このスクリプトを実行すると、Sybase フェールオーバのシステム・プロシージャがインストールされます。

注意 ライセンス情報と enable HA の実行値は互いに独立しています。Sybase フェールオーバのライセンスを取得しているかどうかにかかわらず、実行値と設定値は Adaptive Server の再起動後は 1 に設定されます。ライセンスを取得しなければ、Sybase フェールオーバを実行することはできません。有効なライセンスがインストールされていない場合は、Adaptive Server のログにエラー・メッセージが記録され、この機能はアクティブ化されません。ライセンス・キーのインストールについては、使用しているプラットフォームの『インストール・ガイド』を参照してください。

---

enable HA を 1 または 2 に設定するだけでは、Adaptive Server が高可用性システムで動作するように設定したことにはなりません。『高可用性システムにおける Sybase フェールオーバの使用』で説明する手順を実行して、Adaptive Server が高可用性システムのコンパニオン・サーバになるように設定してください。

enable HA が 0 に設定されているときは、Sybase のフェールオーバ機能に関する設定を行うことはできません。また、*installhasvss* (Windows では *insthasv*) は実行できません。

enable housekeeper GC

要約	
デフォルト値	1 (オン)
値の範囲	0 ～ 5
ステータス	動的
表示レベル	中間
必要な役割	システム管理者
設定グループ	SQL Server 管理

ハウスキーピング・ガーベジ・コレクション・タスクは、データオンリーロック・テーブルの領域を再利用するための処理を実行します。ユーザ・タスクがデータオンリーロック・テーブルからローを削除すると、データ・ページとインデックス・ページにコミットされた削除があるかどうかを調べるハウスキーピングのタスクがキューイングされます。

ハウスキーピング・ガーベジ・コレクション・タスクを制御するには、enable housekeeper GC を使用します。『パフォーマンス＆チューニング・シリーズ：基本』の「第 3 章 エンジンと CPU の使用方法」を参照してください。

enable housekeeper GC の有効値を次に示します。

- 0 － ハウスキーピング・ガーベジ・コレクション・タスクは実行しませんが、delete コマンドによるレイジー・ガーベジ・コレクションは実行できるようにします。reorg reclaim_space を使用して、空ページの割り付けを解除する必要があります。これは、パフォーマンスへの影響が最も少なく、最も低コストのオプションですが、累積した空ページの量が増えるとパフォーマンス上の問題が発生する可能性があります。この値を使用することはおすすめしません。
- 1 － ハウスキーピング・ガーベジ・コレクション・タスクと delete コマンドで、レイジー・ガーベジ・コレクションを実行できます。アプリケーションで許容される以上の空ページが累積する場合は、オプション 4 または 5 の使用を検討してください。optdiag ユーティリティを使用すると、空ページの統計情報を取得できます。
- 2 － 今後のために予約済み。
- 3 － 今後のために予約済み。
- 4 － ハウスキーピング・ガーベジ・コレクション・タスクと delete コマンドで、積極的ガーベジ・コレクションを実行できます。このオプションを選択すれば効果が最も高くなりますが、delete コマンドのコストが高くなります。このオプションは、DOL テーブルに対する一連の削除を 1 つのバッチで実行する場合に理想的です。
- 5 － ハウスキーピング・タスクでは積極的ガーベジ・コレクションを実行でき、delete コマンドではレイジー・ガーベジ・コレクションを実行できます。オプション 4 を選択した場合よりも、削除のコストは低くなります。このオプションは、同時トランザクションによって削除が行われる場合に適しています。

sp_sysmon は、ハウスキーピング・ガーベジ・コレクション・タスクによる領域再利用処理の実行頻度と、再利用可能となったページ数をレポートします。『パフォーマンス＆チューニング・シリーズ：sp_sysmon による Adaptive Server の監視』を参照してください。

enable i/o fencing

要約	
デフォルト値	0
有効な値	1 (有効)、0 (無効)
ステータス	静的
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	共有ディスク・クラスタ

enable i/o fencing を 1 に設定すると、SCSI-3 PGR (Persistent Group Reservation) 規格をサポートする各データベース・デバイスで I/O フェンシング機能が有効になります。

enable java

要約	
デフォルト値	0 (無効)
値の範囲	0 (無効)、1 (有効)
ステータス	静的
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	Java サービス

enable java は、Adaptive Server データベースで Java を使用できるかどうかを指定します。サーバでの Java の使用を有効にしなければ、Java クラスをインストールしたり Java の操作を実行したりすることはできません。

enable job scheduler

要約	
デフォルト値	0 (オフ)
値の範囲	0 (オフ)、1 (オン)
ステータス	動的
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	SQL Server 管理

enable job scheduler は、Adaptive Server の起動時に Job Scheduler を起動するかどうかを指定します。

enable ldap user auth

要約	
デフォルト値	0 (オフ)
有効な値	0 (オフ) − syslogins 認証のみを許可 1 (オン) − LDAP 認証と syslogins 認証の両方を許可 2 (オン) − LDAP 認証のみを許可
ステータス	動的
表示レベル	包括
必要な役割	システム・セキュリティ担当者
設定グループ	セキュリティ関連

enable ldap user auth が 1 に設定されているときは、Adaptive Server は各ユーザを認証するために LDAP サーバを検索します。LDAP 認証に失敗したときは、syslogins を検索してユーザを認証します。レベル 1 は、Adaptive Server 認証から LDAP 認証へユーザをマイグレートしているときに使用します。

enable literal autoparam

要約	
デフォルト値	0
値の範囲	1 (有効)、0 (無効)
ステータス	動的
表示レベル	中間
必要な役割	システム管理者
設定グループ	クエリ・チューニング

enable literal autoparam は、サーバ全体でリテラルの自動パラメータ化を有効または無効にします。

enable logins during recovery

要約	
デフォルト値	1
値の範囲	1 (有効)、0 (無効)
ステータス	動的
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	セキュリティ関連

enable logins during recovery は、データベース・リカバリ中にシステム管理者以外のログインを許可するかどうかを指定します。値 1 はリカバリ中にログインが許可されることを示し、値 0 はリカバリ中にログインが許可されない (システム管理者のみが Adaptive Server にログインできる) ことを示します。

enable merge join

要約	
デフォルト値	2
値の範囲	0 − サーバ・レベルのマージ・ジョインを無効にする 1 − サーバ・レベルのマージ・ジョインを有効にする 2 − サーバ・レベルのマージ・ジョインをデフォルト値に設定する
ステータス	動的
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	クエリ・チューニング

enable merge join サーバ・レベルでマージ・ジョインを有効または無効にします。

マージ・ジョインのデフォルト値は、optimization goal 設定パラメータの現在の値によって異なります。

最適化目標の値	マージ・ジョインのデフォルト値
allrows_mix	on
allrows_dss	on
allrows_oltp	off

enable metrics capture

要約	
デフォルト値	0 (オフ)
値の範囲	0 (オフ)、1 (オン)
ステータス	動的
表示レベル	中間
必要な役割	システム管理者
設定グループ	SQL Server 管理

enable metrics capture は、Adaptive Server が測定基準をサーバ・レベルで取得できるようにします。アドホック文の測定基準はシステム・カタログ内に取得され、ストアド・プロシージャに含まれる文の測定基準はプロシージャ・キャッシュに保存されます。

enable monitoring

要約	
デフォルト値	0 (オフ)
値の範囲	0 (オフ)、1 (オン)
ステータス	動的
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	モニタリング

enable monitoring は、Adaptive Server でモニタリング・テーブル・データを収集するかどうかを制御します。enable monitoring は、Monitoring 設定パラメータが有効になるかどうかを指定するマスタ・スイッチの働きをします。

enable pam user auth

要約	
デフォルト値	0 (オフ)
値の範囲	0 (オフ) − syslogins 認証のみを許可 1 (オン) − PAM 認証と syslogins 認証の両方を許可 2 (オン) − PAM 認証のみを許可
ステータス	動的
表示レベル	中間
必要な役割	システム・セキュリティ担当者
設定グループ	セキュリティ関連

enable pam user auth は、PAM (Pluggable Authentication Modules) を使用してユーザを管理する能力を制御します。

enable pam user auth が 1 に設定されているときは、Adaptive Server は各ユーザを認証するために PAM プロバイダを使用します。PAM 認証に失敗したときは、syslogins を検索してユーザを認証します。レベル 1 は、Adaptive Server 認証から PAM 認証へユーザをマイグレートしているときに使用します。

enable pci

要約	
デフォルト値	0 (オフ)
有効な値	0 (オフ)、1 (オン)、2 (オペレーティング・システムの無効化によりオン)
ステータス	動的
表示レベル	中間
必要な役割	システム管理者
設定グループ	ユーザ環境

enable pci は、Adaptive Server の Java PCI Bridge を有効または無効にします。

注意 Sybase 製品の保守契約を結んでいるサポート・センタの指示がないかぎり、設定 "2" (オペレーティング・システムの無効化によりオン) を使用しないでください。この設定を使用すると、PCI 機能を完全または正しくサポートしていない可能性があるオペレーティング・システムのバージョンにおいても PCI Bridge が有効になります。

enable query tuning mem limit

要約	
デフォルト値	1 (オン)
値の範囲	0 (オフ)、1 (オン)
ステータス	動的
表示レベル	中間
必要な役割	システム管理者
設定グループ	クエリ・チューニング

enable query tuning mem limit は、クエリ・チューニングのメモリ制限を有効にします。

enable query tuning time limit

要約	
デフォルト値	1 (オン)
値の範囲	0 (オフ)、1 (オン)
ステータス	中間
表示レベル	中間
必要な役割	システム管理者
設定グループ	クエリ・チューニング

enable query tuning time limit は、クエリ・チューニングの時間制限を有効にします。

enable real time messaging

要約	
デフォルト値	1 (オン)
値の範囲	0 (オフ)、1 (オン)
ステータス	動的
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	SQL Server 管理

enable real time messaging は、Real Time Messaging Services を有効にします。

enable rep agent threads

要約	
デフォルト値	1 (オン)
値の範囲	0 (オフ)、1 (オン)
ステータス	動的
表示レベル	基本
必要な役割	システム管理者
設定グループ	メモリ使用、RepAgent スレッド管理

enable rep agent threads パラメータは、Adaptive Server 内で RepAgent スレッドを実行できるようにします。

複写を使用できるようにするには、他の手順も実行する必要があります。詳細については、Replication Server のマニュアルを参照してください。

enable row level access control

要約	
デフォルト値	0 (オフ)
有効な値	0 (オフ)、1 (オン)
ステータス	動的
表示レベル	包括
必要な役割	システム・セキュリティ担当者
設定グループ	セキュリティ関連

enable row level access control は、ロー・レベルのアクセス制御を有効にします。enable row level access control を設定するためには、事前にセキュリティ・サービス・ライセンスを有効にしておくことが必要です。

enable semantic partitioning

要約	
デフォルト値	0
値の範囲	1 (有効)、0 (無効)
ステータス	動的
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	SQL Server 管理

enable semantic partitioning は、Adaptive Server でラウンドロビン方式以外の分割 (リスト、ハッシュ、範囲による分割) を実行できるようにします。これらの分割スキームを使用するには、適切なライセンスを所有している必要があります。

enable sort-merge join and JTC

要約	
デフォルト値	0 (オフ)
有効な値	0 (オフ)、1 (オン)
ステータス	動的
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	クエリ・チューニング

Adaptive Server が互換モードのときに使用されます。いったん有効にされると、Adaptive Server がクエリを互換モードでコンパイルすると、enable sort-merge join and JTC はクエリ・プロセッサがソート・マージ・ジョインまたはネストされたループ・ジョインを選択できるようにします。enable sort-merge join and JTC は、JTC (Join Transitive Closure) を有効にします。これにより、15.0 より前のリリースのクエリ・プロセッサは追加のジョイン句を使用できるようになります。

互換モードの詳細については、『マイグレーション技術ガイド』を参照してください。

enable sql debugger

要約	
デフォルト値	1 (オン)
有効な値	0 (オフ)、1 (オン)
ステータス	静的
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	SQL Server 管理

Adaptive Server SQL デバッガを有効または無効にします。このデバッガを使用すると、T-SQL コードを 1 ステップずつ実行できます。

enable ssl

要約	
デフォルト値	0 (オフ)
有効な値	0 (オフ)、1 (オン)
ステータス	静的
表示レベル	包括
必要な役割	システム・セキュリティ担当者
設定グループ	セキュリティ関連

enable ssl は、Secure Sockets Layer セッションベースのセキュリティを有効または無効にします。

enable stmt cache monitoring

要約	
デフォルト値	0 (オフ)
有効な値	0 (オフ)、1 (オン)
ステータス	動的
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	モニタリング

enable stmt cache monitoring は、Adaptive Server がステートメント・キャッシュに関するモニタリング情報を収集する機能を有効または無効にします。いったん有効にされると、monStatementCache と monCachedStatement は有効なデータを表示します。

enable surrogate processing

要約	
デフォルト値	1 (オン)
値の範囲	0 (オフ)、1 (オン)
ステータス	動的
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	Unicode

サロゲート処理をアクティブにし、Unicode データのサロゲート・ペアが整合性を保つようにします。enable surrogate processing を無効にすると、Unicode データ内のサロゲート・ペアは無視され、サロゲート・ペアの整合性を維持するコードがすべてスキップされます。これによってパフォーマンスは向上しますが、データとして表示される Unicode 文字の範囲は小さくなります。

enable unicode conversion

要約	
デフォルト値	1
値の範囲	0 − 組み込みの文字セット変換のみ使用。 1 − 組み込み変換を使用。組み込みの文字セット変換が見つからないときは、Unilib 文字変換を使用する。 2 − 適切な Unilib 変換を使用。
ステータス	動的
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	言語、Unicode

enable unicode conversion は、char、varchar、text の各データ型について、Unilib を使用した文字変換をアクティブにします。

enable unicode normalization

要約	
デフォルト値	1 (オン)
値の範囲	0 (オフ)、1 (オン)
ステータス	動的
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	Unicode

Unilib 文字正規化をアクティブにします。正規化とは、特定の抽象文字シーケンスに対する表現がデータベース内に 1 つだけ存在するようにデータを修正するプロセスのことです。多くの場合、発音区別符号が後に付いた文字が、事前結合済みの形式に置き換えられます。

入力されたすべての Unicode データに対して正規化を実行する組み込みプロセスを使用する場合は、enable unicode normalization を 1 に設定します。このパラメータを無効にする (0 に設定する) と、正規化が省略されるため、サーバではなくクライアント・コード側で正規化を行うことになります。正規化を無効にするとパフォーマンスが向上します。ただし、すべてのクライアントが同じ表現を使用して Unicode データをサーバに渡す場合に限られます。

注意　いったん無効にしても、再び有効にできます。これにより、正規化されていないデータがデータベースに格納されることが防止されます。

enable webservices

要約	
デフォルト値	0
値の範囲	1 (有効)、0 (無効)
ステータス	動的
表示レベル	中間
必要な役割	システム管理者
設定グループ	SQL Server 管理

Web Services を有効にします。

enable xact coordination

要約	
デフォルト値	1 (オン)
有効な値	0 (オフ)、1 (オン)
ステータス	静的
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	DTM 管理

enable xact coordination は、Adaptive Server トランザクション・コーディネーション・サービスを有効または無効にします。このパラメータを 1 (オン) に設定するとコーディネーション・サービスが有効になり、サーバは他の Adaptive Server にトランザクションを送信できます。これが発生するのは、トランザクションがリモート・プロシージャ・コール (RPC) を実行して他のサーバのデータを更新するか、コンポーネント統合サービス (CIS) を使用して他のサーバのデータを更新する場合です。トランザクション・コーディネーション・サービスは、リモート・サーバのデータへの更新が、必ずオリジナル・トランザクションとともにコミットまたはロールバックされるようにする機能です。

このパラメータを 0 (オフ) に設定すると、Adaptive Server はリモート・サーバの作業をコーディネートしません。トランザクションで RPC を実行することや CIS を使用してデータを更新することはできますが、リモート・サーバにシステム障害が発生したときに、リモート・トランザクションがオリジナル・トランザクションとともにロールバックされることや、リモートの作業がオリジナル・トランザクションとともにコミットされることを Adaptive Server が保証することはできません。これは、バージョン 12.x. より前の Adaptive Server の動作に対応しています。

enable xml

要約	
デフォルト値	0
値の範囲	1 (有効)、0 (無効)
ステータス	動的
表示レベル	中間
必要な役割	システム管理者
設定グループ	SQL Server 管理

XML サービスを有効にします。

engine memory log size

要約	
デフォルト値	0
値の範囲	0 ～ 2147483647
ステータス	動的
表示レベル	
必要な役割	
設定グループ	物理メモリ

engine memory log size は、診断専用であり、運用環境には関連しません。Sybase 製品の保守契約を結んでいるサポート・センタからの指示がないかぎり、デフォルト値のままにしておいてください。

errorlog pipe active

要約	
デフォルト値	0
値の範囲	0 ～ 1
ステータス	動的
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	モニタリング

errorlog pipe active は、Adaptive Server でエラー・ログ・メッセージを収集するかどうかを制御します。errorlog pipe active と errorlog pipe max messages を両方とも有効にすると、Adaptive Server はエラー・ログに送られたすべてのメッセージを収集します。このエラー・ログ・メッセージを取得するには、monErrorLog を使用します。

errorlog pipe max messages

要約	
デフォルト値	0
値の範囲	0 ～ 2147483647
ステータス	動的
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	メモリ使用、モニタリング

errorlog pipe max messages は、Adaptive Server が格納するエラー・ログのメッセージ数をエンジンごとに決定します。monSQLText テーブル内のメッセージ数の合計は、sql text pipe max messages に実行中のエンジン数を掛け合わせた値になります。

esp execution priority

要約	
デフォルト値	8
値の範囲	0 ～ 15
ステータス	動的
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	拡張ストアド・プロシージャ

esp execution priority は、ESP 実行用の XP Server スレッドの優先度を設定します。ESP は、長時間にわたって CPU を集中的に使用することがあります。また、XP Server は Adaptive Server と同じマシン上に常駐するため、Adaptive Server のパフォーマンスにも影響を与えることがあります。

Open Server スレッドのスケジューリングについては、『Open Server Server-Library/C リファレンス・マニュアル』を参照してください。

esp execution stacksize

要約	
デフォルト値	34816
値の範囲	34816 ～ 2^{14}
ステータス	静的
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	拡張ストアド・プロシージャ

esp execution stacksize は、ESP 実行用に割り付けるスタック・サイズをバイト単位で設定します。

デフォルトの 34816 よりも大きいスタック･サイズを必要とする独自の ESP 関数を使用する場合に、このパラメータを使用します。

esp unload dll

要約	
デフォルト値	0 (オフ)
値の範囲	0 (オフ)、1 (オン)
ステータス	動的
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	拡張ストアド・プロシージャ

esp unload dll は、ESP の呼び出し完了後に、ESP をサポートする DLL を XP Server のメモリから自動的にアンロードするかどうかを指定します。

esp unload dll を 0 に設定すると、DLL は自動的にはアンロードされません。1 に設定すると、自動的にアンロードされます。

esp unload dll を 0 に設定した場合でも、sp_freedll を使用すれば、実行時に個々の DLL を明示的にアンロードできます。

event buffers per engine

要約	
デフォルト値	100
値の範囲	1 ～ 2147483647
ステータス	静的
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	メモリ使用、SQL Server 管理

event buffers per engine は、Adaptive Server Monitor で同時にモニタできる Adaptive Server エンジン当たりのイベント数を指定します。イベントは、Adaptive Server のパフォーマンスを監視するために Adaptive Server Monitor が使用します。Adaptive Server Monitor を使用していない場合は、このパラメータを 1 に設定してください。

event buffers per engine に設定する値は、設定されているエンジン数、Adaptive Server のアクティビティのレベル、実行するアプリケーションの種類によって決まります。

event buffers per engine を小さな値に設定すると、イベント情報が失われることがあります。デフォルト値は、ほとんどのサイトにとって小さすぎる可能性があります。一般的なモニタには、2,000 以上の値が妥当です。ただし、サイトにとって適切な値を決めるには経験が必要です。

一般に、event buffers per engine の設定値を大きくすれば、Adaptive Server Monitor が原因で起きる Adaptive Server のパフォーマンス低下を緩和することができます。

それぞれのイベント・バッファは 100 バイトのメモリを使用します。event buffers per engine に特定の値を設定した場合に使用されるメモリの合計量を調べるには、event buffers per engine の値に、設定されている Adaptive Server エンジン数を掛けます。

event log computer name (Windows のみ)

要約	
デフォルト値	LocalSystem
有効な値	• Adaptive Server のメッセージを記録するように設定されている、ネットワーク上の Windows マシンの名前 • LocalSystem • NULL
ステータス	動的
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	エラー・ログ

event log computer name には、Adaptive Server のメッセージを Windows のイベント・ログに記録する Windows PC マシンの名前を指定します。この機能は、Windows サーバでのみ使用できます。

LocalSystem または NULL を設定すると、デフォルトのローカル・システムが指定されます。

また、Server Config ユーティリティを使用して、event log computer name パラメータを設定することもできます。その場合は、[イベント・ログ] でコンピュータ名を指定します。

コマンド・ラインで -G オプションを指定した場合に、sp_configure で event log computer name パラメータを設定するか、[イベント・ログ] でコンピュータ名を指定すると、このオプションの結果は上書きされます。-G オプションを指定して Adaptive Server を起動した場合は、event log computer name パラメータを設定することによって送信先リモート・マシンを変更できます。

Adaptive Server メッセージをリモート・サイトでログに記録する方法の詳細については、『Adaptive Server Enterprise 設定ガイド Windows 版』を参照してください。

event logging (Windows のみ)

要約	
デフォルト値	1
有効な値	0 (オフ)、1 (オン)
ステータス	動的
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	エラー・ログ

event logging は、Windows のイベント・ログに Adaptive Server のメッセージを記録するかどうかを指定します。

デフォルト値の 1 を指定すると Adaptive Server のメッセージは Windows のイベント・ログに記録され、0 を指定するとイベント・ログには記録されません。

Server Config ユーティリティを使用して event logging パラメータを設定するには、[イベント・ログ] の [Windows イベント・ログを使用] を選択します。

コマンド・ラインで -G オプションを指定した場合に、event logging パラメータを設定するか、[Windows イベント・ログを使用] を選択すると、このオプションの結果は上書きされます。

executable codesize + overhead

要約	
デフォルト値	0
値の範囲	0 ～ 2147483647
ステータス	計算された値
表示レベル	基本
必要な役割	システム管理者
設定グループ	メモリ使用

executable codesize + overhead は、Adaptive Server の実行プログラムとオーバヘッドを合わせたサイズをキロバイト単位で表します。これは、システムによって計算される値であって、ユーザが設定できるものではありません。

extended cache size

要約	
デフォルト値	0
値の範囲	0 ～ 31457280
ステータス	静的
表示レベル	中間
必要な役割	システム管理者
設定グループ	キャッシュ・マネージャ

extended cache size は、セカンダリ・キャッシュのサイズを指定します。

FIPS login password encryption

要約	
デフォルト値	0
値の範囲	0 ～ 1
ステータス	静的
表示レベル	包括
必要な役割	システム・セキュリティ担当者
設定グループ	セキュリティ関連

FIPS login password encryption を有効にするには、セキュリティ&ディレクトリサービス ライセンスが必要です。このパラメータは、FIPS 140-2 暗号化モジュールをサポートして、転送時、メモリ内、ディスク上でのパスワードを暗号化します。

Adaptive Server では、ログイン暗号化に FIPS 140-2 認定の Certicom 製セキュリティ・プロバイダが使用されます。この設定が有効でない場合、Adaptive Server は OpenSSL セキュリティ・プロバイダを使用してログイン・パスワードの暗号化を実行します。

global async prefetch limit

要約	
デフォルト値	10
値の範囲	0 ～ 100
ステータス	動的
表示レベル	中間
必要な役割	システム管理者
設定グループ	キャッシュ・マネージャ

global async prefetch limit は、非同期プリフェッチによってバッファ・プールに取り込まれたけれどもまだ読み込まれていないページを保持できる割合を指定します。このパラメータは、制限値が sp_poolconfig で明示的に設定されていないすべてのキャッシュ内のすべてのプールの制限値を設定します。

プールの制限値を超えた場合、読み込まれていないページの割合が制限値より小さくなるまで、非同期プリフェッチは一時的に無効になります。『パフォーマンス&チューニング・シリーズ：基本』の「第 6 章 非同期プリフェッチのチューニング」を参照してください。

global cache partition number

要約	
デフォルト値	1
値の範囲	1 ～ 64 (2 の累乗)
ステータス	静的
表示レベル	中間
必要な役割	システム管理者
設定グループ	キャッシュ・マネージャ

global cache partition number は、すべてのデータ・キャッシュのキャッシュ・パーティションのデフォルト数を設定します。特定のキャッシュのパーティション数は、sp_cacheconfig を使用して設定できます。ローカルのパーティション数がグローバルのパーティション数よりも優先します。

キャッシュ・パーティションを使用するとキャッシュ・スピンロックの競合が減少します。一般に、キャッシュ・スピンロックの競合が 10% を超えるときは、キャッシュを分割することでパフォーマンスが向上します。パーティションの数を 2 倍にすると、スピンロックの競合がおよそ 2 分の 1 に減少します。

キャッシュ・パーティションの設定の詳細については、『システム管理ガイド第 2 巻』の「第 4 章 データ・キャッシュの設定」を参照してください。また、『パフォーマンス&チューニング・シリーズ：基本』の「第 6 章 非同期プリフェッチのチューニング」も参照してください。

heap memory per user

要約	
デフォルト値	4K
有効な値	0 ～ 2147483647 バイト
ステータス	動的
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	メモリ使用、物理メモリ

heap memory per user は、ユーザ当たりのヒープ・メモリ量を設定します。ヒープ・メモリ・プールは起動時に作成される内部メモリで、このプールからタスクが必要に応じて動的にメモリを割り付けます。このメモリ・プールが重要な役割を果たすのは、ワイド・カラムを使用するため大量のメモリを必要とするタスクを実行する場合です。ヒープ・メモリからテンポラリ・バッファが割り付けられることによって、ワイド・カラムを使用するタスクの実行が可能となります。タスクが使用するヒープ・メモリは、タスクが完了するとヒープ・メモリ・プールに返されます。

メモリ・プールのサイズは、ユーザ接続数によって異なります。heap memory per user は、論理ページ・サイズの 3 倍に設定することをおすすめします。

histogram tuning factor

要約	
デフォルト値	20
値の範囲	1 ～ 100
ステータス	動的
表示レベル	中間
必要な役割	システム管理者
設定グループ	SQL Server 管理

histogram tuning factor は、Adaptive Server が update statistics、update index statistics、update all statistics、create index について 1 つのヒストグラムで分析するステップ数を制御します。値を 1 に設定すると、パラメータが無効になります。

注意 Adaptive Server バージョン 15.0.2 ESD #2 およびそれ以降の場合、histogram tuning factor をデフォルト値の 20 に設定し、しかもヒストグラムに多数のステップが必要とされる場合、ヒストグラムに使用される実際のステップ数は、プロシージャ・キャッシュ使用量を減らす値に制限されます。

```
min (max (400, requested_steps),
histogram_tuning_factor X requested_steps)
```

次の例では、Adaptive Server は 30 個の値を含む 20 ステップの中間ヒストグラムを生成します。

```
sp_configure 'histogram tuning factor',20
update statistics tab using 30 values
```

Adaptive Server は、次の条件に従って、ヒストグラムを分析して結果のヒストグラムに圧縮します。

- 最初のステップは変更せずにコピーする。
- 頻度の高いステップは変更せずにコピーする。
- 連続した範囲のステップは 1 つの結果ステップにまとめる。このため、まとめられたステップの総ウェイトは値の 30 分の 1 を超えない。

sysstatistics 内の最終的なヒストグラムは次のようになります。

- 範囲ステップは 30 ステップの update statistics と同様に生成され、高頻度の範囲はヒストグラムが 600 ステップで作成された場合のように分離されます。
- 結果ヒストグラムの合計ステップ数は、30 ～ 600 の値になります。
- 均等に分散したデータの場合、値は 30 にかぎりなく近づきます。
- テーブルでの「頻度」が高い値は、ヒストグラムでのステップ数が多くなります。
- 1 つのカラムに異なる値がわずかしかない場合は、それらすべての値が高頻度セルとして表示されることがあります。

number of histogram steps を 600 に増やしても同じ結果が得られますが、バッファやプロシージャ・キャッシュでより多くのリソースを使用することになります。

histogram tuning factor を使用すると、ヒストグラムで使用されるリソースを最小限に抑えられます。リソース使用量が増えるのは、最適化のために適切な場合のみです。たとえば、カラムのデータ分布に一貫性がない場合、または重複する値がカラム内に多数存在する場合です。このような場合には、最大 600 のヒストグラム・ステップが使用されます。ただし、ほとんどの場合、histogram tuning factor はデフォルト値 (上記の例では 30) が使用されます。

housekeeper free write percent

要約	
デフォルト値	1
値の範囲	0 ～ 100
ステータス	動的
表示レベル	中間
必要な役割	システム管理者
設定グループ	SQL Server 管理

housekeeper free write percent パラメータは、ハウスキーピング・ウォッシュ・タスクによるデータベースへの書き込みの最大増加率を指定します。

たとえば、データベースへの書き込み頻度が通常より 5 パーセント高くなった場合にハウスキーピング・タスクの動作を停止させるには、housekeeper free write percent を 5 に設定します。

Adaptive Server が処理するユーザ・タスクがなくなると、ハウスキーピング・ウォッシュ・タスクは、変更されたページをキャッシュからディスクに書き込む処理を自動的に開始します。この書き込みによって CPU 使用率が改善され、トランザクション処理中のバッファ・ウォッシングの必要性が少なくなり、チェックポイントが短くなります。

同じデータベース・ページを繰り返し更新するアプリケーションでは、ハウスキーピング・ウォッシュによるデータベースへの書き込みの中には実際には不必要なものもあります。このような書き込みはサーバのアイドル時間中にだけ発生しますが、ディスクへの負荷が大きいシステムでは許容できないことがあります。

クエリを最適化するために使用されるテーブルとインデックス統計は、クエリ処理中はメモリ構造内に保持されます。この統計が変化しても、変更内容はすぐには systabstats テーブルに書き込まれません。これは、I/O 競合を削減し、パフォーマンスを向上させるためです。代わりに、ハウスキーピング・チョア・タスクによって統計が定期的にディスクにフラッシュされます。

デフォルト値を使用する場合は、ディスク I/O が最大 1 パーセント増加するまでハウスキーピング・ウォッシュ・タスクを実行できます。これは、ほとんどのシステムでパフォーマンスとリカバリ・スピードを改善します。

ハウスキーピング・ウォッシュ・タスクが実行されないようにするには、housekeeper free write percent の値を 0 に設定します。

システムでのディスク競合が多く、ハウスキーピング・ウォッシュ・タスクによる I/O の増加が許容されない場合にのみ、この値を 0 に設定します。

ハウスキーピングを無効にした場合は、統計情報を常に最新の状態に保ってください。ディスクに統計を書き込むコマンドは、次のとおりです。

- update statistics
- dbcc checkdb (データベース内のすべてのテーブルの場合) または dbcc checktable (1 つのテーブルの場合)
- sp_flushstats

統計がディスクに最後に書き込まれた後で更新されたテーブルに対して、これらのコマンドのいずれかを、次の時点で実行してください。

- データベースのダンプの前
- 正常なシャットダウンの前
- 失敗または正常なシャットダウンの後の再起動の後。このような場合、sp_flushstats は使用できません。update statistics コマンドまたは dbcc コマンドを使用してください。
- テーブルへの大幅な変更後。たとえば、大量のバルク・コピー・オペレーション、ロック・スキームの変更、大量のローの削除や挿入、truncate table コマンドの実行などがあります。

データベース書き込みの増加率に関係なくハウスキーピング・ウォッシュ・タスクが連続的に動作するようにするには、housekeeper free write percent を 100 に設定します。

ハウスキーピングのパフォーマンスをモニタリングするには、sp_sysmon を使用します。『パフォーマンス&チューニング・シリーズ：sp_sysmon による Adaptive Server の監視』を参照してください。

また、ハウスキーピング・タスクによって発生したフリー・チェックポイントの数を調べることもできます。この出力については、『パフォーマンス&チューニング・シリーズ：基本』を参照してください。

i/o accounting flush interval

要約	
デフォルト値	1000
値の範囲	1 ～ 2147483647
ステータス	動的
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	SQL Server 管理

i/o accounting flush interval は、Adaptive Server が各ユーザについての I/O 統計情報を sysprocesses から syslogins にフラッシュするまでの待ち時間をマシンのクロック・チック単位で指定します。これはチャージバック・アカウンティングに使用します。

ユーザが Adaptive Server にログインすると、その時点以降のユーザ・プロセスの I/O 統計情報が sysprocesses に蓄積されます。i/o accounting statistics interval の時間が過ぎるか、ユーザが Adaptive Server からログオフすると、蓄積された I/O 統計情報が sysprocesses から syslogins にフラッシュされます。この統計情報は、システム管理者が sp_clearstats を使用して合計をクリアするまでは、syslogins に引き続き蓄積されます。syslogins から現在の合計を表示するには、sp_reportstats を使用します。

i/o accounting flush interval に設定する値は、目的とするレポートのタイプによって異なります。月単位でレポートを作成する場合は、i/o accounting flush interval には比較的大きな値を設定します。レポート処理の頻度が低ければ、syslogins 内のデータを頻繁に更新することはそれほど重要ではありません。

プロセスによる I/O 量を調べるために、アドホック・クエリを使用して syslogins の totio カラムからの選択を定期的に実行する場合は、i/o accounting flush interval の設定値を小さくします。このようにすれば、選択を実行するときに syslogins 内のデータが最新のものである可能性が高くなります。

I/O 統計情報をレポートしない場合は、i/o accounting flush interval を最大値に設定してください。これにより、syslogins が更新される回数と、そのページをディスクに書き込まなければならない回数が減ります。

i/o batch size

要約	
デフォルト値	100
値の範囲	1 ～ 2147483647
ステータス	動的
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	SQL Server 管理

i/o batch size は、タスクがスリープする前にバッチ内で発行される書き込みの回数を設定します。バッチが完了すると、タスクがウェイクアップして、次の書き込みバッチが発行されます。これにより、I/O サブシステムに対して大量の書き込みが同時に発生することを防止できます。i/o batch size に適切な値を設定することで、checkpoint や dump database、select into などの操作のパフォーマンスを向上させることができます。

i/o polling process count

要約	
デフォルト値	10
値の範囲	1 ～ 2147483647
ステータス	動的
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	SQL Server 管理

i/o polling process count は、Adaptive Server が実行可能なプロセスの最大数を指定します。この数に達したとき、スケジューラはディスク I/O やネットワーク I/O が完了しているかどうかを調べます。i/o polling process count を調整すると、Adaptive Server の応答時間とスループットの両方が影響を受けます。

Adaptive Server は、次のときにディスク I/O またはネットワーク I/O が完了しているかどうかを調べます。

- 前回 Adaptive Server が I/O の完了を調べた後に実行されたタスク数が、i/o polling process count の値に等しくなったとき
- Adaptive Server のクロック・チックごと

一般的には、i/o polling process count の値を増やすと、大量のディスク I/O とネットワーク I/O を行うアプリケーションのスループットが向上します。逆にこの値を減らすと、スループットが低下するリスクはありますが、このような。

アプリケーションが I/O バウンド・タスクと CPU バウンド・タスクの両方を作成する場合は、i/o polling process count を 1 ～ 2 の小さな値に調整すると、I/O バウンド・タスクが確実に CPU サイクルにアクセスできるようになります。

OLTP アプリケーション (またはユーザ接続と短いトランザクションを使用する I/O バウンド・アプリケーション) の場合は、i/o polling process count を 20 ～ 30 の範囲の値に調整すると、スループットが増加することがありますが、応答時間が長くなる可能性もあります。

i/o polling process count を調整する場合は、次の 3 つのパラメータを考慮してください。

- sql server clock tick length。このパラメータは、Adaptive Server のクロック・チックをマイクロ秒単位で指定します。“sql server clock tick length” (232 ページ) を参照してください。
- time slice。このパラメータは、Adaptive Server のスケジューラがユーザ・プロセスの実行を許容する時間をクロック・チック数として指定します。“time slice” (246 ページ) を参照してください。
- cpu grace time。このパラメータは、ユーザ・プロセスが CPU を解放せずに実行できる時間の最大長をクロック・チック単位で指定します。この時間に達すると、Adaptive Server はユーザ・プロセスの制御を横取りしてタイムスライス・エラーで終了させます。“cpu grace time” (97 ページ) を参照してください。

i/o polling process count の変更の効果を調べるには、sp_sysmon を使用します。『パフォーマンス＆チューニング・シリーズ：sp_sysmon による Adaptive Server の監視』を参照してください。

identity burning set factor

要約	
デフォルト値	5000
値の範囲	1 ～ 9999999
ステータス	静的
表示レベル	中間
必要な役割	システム管理者
設定グループ	SQL Server 管理

IDENTITY カラムは、numeric 型で位取りがゼロの、Adaptive Server によって値が生成されるカラムです。カラム値の最小値は 1 で、最大値はカラムの精度によって決まります。

Adaptive Server は、IDENTITY カラムのあるテーブルごとに、カラム値として可能な一連の値を連続した番号のブロックに分けて、メモリ内で一度に 1 ブロックずつ使用できるようにします。テーブルにローが挿入されるたびに、そのブロックから次に使用可能な値を IDENTITY カラムに割り当てます。ブロック内のすべての番号を使いきると、次のブロックが使用可能になります。

IDENTITY カラム値を選択するこの方法は、サーバのパフォーマンスを改善します。Adaptive Server は、新しいカラム値を割り当てるときに、メモリから現在の最大値を読み込んで 1 を加えます。ディスク・アクセスが必要になるのは、ブロック内の値を使いきったときだけです。サーバの障害が発生すると (または shutdown with nowait が実行されると)、ブロック内に残っている番号はすべて破棄されるので、この方法では IDENTITY カラム値に欠番が発生することがあります。

identity burning set factor は、それぞれのブロックで使用可能にすることができるカラムの値の割合を変更するために使用します。この値は適切なパフォーマンスを得るのに十分な大きさにしますが、カラム値の欠番が許容できなくなるほどには大きくしないでください。デフォルト値の 5,000 は、IDENTITY カラム値の 0.05% を一度に使用できるようにします。

sp_configure で値を正しく設定するには、パーセンテージを小数で表した値を 10^7 (10,000,000) 倍します。たとえば、IDENTITY カラム値の候補の 15% を一度に使用できるようにするには、0.15×10^7 (つまり 1,500,000) を sp_configure で指定します。

identity grab size

要約	
デフォルト値	1
値の範囲	1 ～ 2147483647
ステータス	動的
表示レベル	中間
必要な役割	システム管理者
設定グループ	SQL Server 管理

identity grab size は、Adaptive Server の各プロセスが、IDENTITY カラムを持つテーブルに挿入する IDENTITY カラム値のブロックを予約できるようにするためのパラメータです。

これは、挿入処理を実行するときに、すべての挿入データの IDENTITY 番号が連続するようにする場合に便利です。たとえば支払給与データの入力中に、特定の部署に関するすべてのレコードを同じローのブロック内に置く場合は、identity grab size をその部署のレコード数に設定します。

identity grab size に設定する値は、Adaptive Server のすべてのユーザに適用されます。このため、identity grab size を大きな値に設定すると、IDENTITY カラムを持つテーブルに多くのユーザがデータを挿入するような場合、IDENTITY カラムに大きな欠番が生じます。

identity grab size は、連続するローとして挿入するレコードのグループの中で最も大きなものに対応できるよう十分に大きな値に設定することをおすすめします。

identity reservation size

要約	
デフォルト値	1
値の範囲	1 ～ 2147483647
ステータス	動的
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	SQL Server 管理

identity reservation size は、identity 値の数に制限を設定します。

idle migration timeout

要約	
デフォルト値	60
有効な値	0 ～ 32767
ステータス	動的
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	共有ディスク・クラスタ

idle migration timeout は、クライアントに送信されたマイグレーション要求を無効にせずにアイドル接続を閉じるまでの時間を指定します。これにより、アイドル状態のクライアント接続がマイグレートするのを待つことなく指定時間後にインスタンスを停止できます。

idle migration timeout を高い値に設定すると、適切な停止が遅くなります。これは、クライアントがマイグレーションを開始することのないマイグレーション要求を発行したアイドル接続すべてについて、インスタンスが指定時間だけ待たなければならないからです。

job scheduler interval

要約	
デフォルト値	1 (分単位)
値の範囲	1 ～ 600
ステータス	動的
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	SQL Server 管理

job scheduler interval は、実行すべきスケジュール・ジョブはどれかを Job Scheduler がチェックする間隔を設定します。

job scheduler tasks

要約	
デフォルト値	32
値の範囲	1 ～ 640
ステータス	動的
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	SQL Server 管理

job scheduler tasks は、Job Scheduler を介して同時に実行できるジョブの最大数を設定します。

license information

要約	
デフォルト値	25
有効な値	0 ～ 2^{31}
ステータス	動的
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	SQL Server 管理

license information パラメータは、Sybase のシステム管理者が Adaptive Server で使用されているユーザ・ライセンスの数をモニタリングできるようにします。このパラメータを有効にした場合でも、発行されたライセンスの数がモニタリングされるだけで、ライセンス契約が強制されることはありません。

license information を 0 に設定すると、Adaptive Server はライセンスの使用をモニタリングしません。license information を 0 より大きな値に設定すると、ハウスキーピング・チョア・タスクが Adaptive Server のアイドル・サイクル中に使用されたライセンスの数をモニタリングします。license information は、使用しているライセンス契約で指定されたライセンスの数に設定してください。

使用されているライセンスの数が license information に設定されている数より大きい場合、Adaptive Server はエラー・ログに次のエラー・メッセージを書き込みます。

```
WARNING: Exceeded configured number of user licenses
```

24 時間ごとに、その時間中に使用されたライセンスの最大数が syblicenseslog テーブルに追加されます。Adaptive Server を再起動すると、この 24 時間の周期がリセットされます。

「ライセンス使用状況のモニタリング」(453 ページ) を参照してください。

lock address spinlock ratio

要約	
デフォルト値	100
値の範囲	1 ～ 2147483647
ステータス	静的
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	ロック・マネージャ

Adaptive Server で複数のエンジンを実行する場合に、address lock spinlock ratio を使用して、1 つのスピンロックによって保護される内部アドレス・ロックのハッシュ・テーブルのローの数を設定します。

Adaptive Server は、1031 のローを持つ内部ハッシュ・テーブル (ハッシュ・バケットと呼ばれる) を使用して、アドレス・ロックの取得と解放を管理します。このテーブルは、1 つまたは複数のスピンロックを使用して、異なるエンジンで実行しているプロセス間のアクセスを直列化できます。

address lock spinlock ratio のデフォルト値により、アドレス・ロックのハッシュ・テーブルに対して 11 個のスピンロックが定義されます。最初の 10 個のスピンロックはそれぞれ 100 のローを保護し、11 番目のスピンロックは残りの 31 のローを保護します。address lock spinlock ratio に 1031 以上の値を指定すると、テーブル全体に対してスピンロックが 1 つだけ使用されます。

lock hashtable size

要約	
デフォルト値	2048
値の範囲	1 ～ 2147483647
ステータス	静的
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	ロック・マネージャ、メモリ使用

lock hashtable size は、ロック・ハッシュ・テーブル内のハッシュ・バケットの数を指定します。このテーブルによって、すべてのロー、ページ、テーブルのロックとロック要求が管理されます。タスクがロックを取得するたびに、ロックはハッシュ・バケットに割り当てられ、そのロックへのロック要求は、該当するハッシュ・バケットをチェックします。この値を小さくすると、各ハッシュ・バケット内のロック数が増え、検索時間が長くなります。複数のエンジンを持つ Adaptive Server では、この設定値が小さすぎると、スピンロックの競合が増加する可能性もあります。この値は、デフォルトの 2048 よりも小さくしないでください。

lock hashtable size は、2 の累乗でなければなりません。指定した値が 2 の累乗でない場合には、sp_configure は次に大きい 2 の累乗に値を切り上げ、情報メッセージを表示します。

最適なハッシュ・テーブル・サイズは、同時にロックできる個別のオブジェクト (ページ、テーブル、ロー) の数と相関関係があります。ハッシュ・テーブル・サイズは、同時にロックしなければならない個別のオブジェクト数の少なくとも 20 パーセントが最適です。詳細については、『パフォーマンス&チューニング・シリーズ：ロックと同時実行制御』を参照してください。

lock scheme

要約	
デフォルト値	allpages
値の範囲	allpages、datapages、datarows
ステータス	動的
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	ロック・マネージャ

lock scheme は、create table と select into でロック・スキームが指定されていない場合に使用するデフォルトのロック・スキームを設定します。

ロック・スキームの値は文字データです。そのため、数値でなければならない 2 番目のパラメータのプレースホルダとして 0 を使用する必要があります。3 番目のパラメータに、allpages、datapages、datarows のいずれかを指定します。

```
sp_configure "lock scheme", 0, datapages
```

lock shared memory

要約	
デフォルト値	0 (オフ)
有効な値	0 (オフ)、1 (オン)
ステータス	静的
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	物理メモリ

lock shared memory は、Adaptive Server のページをディスクにスワップさせないようにし、オペレーティング・システム・カーネルがサーバの内部ページ・ロック・コードを回避できるようにします。これにより、コストのかかるディスクの読み込みを減らせます。

すべてのプラットフォームが、共有メモリのロックをサポートしているわけではありません。プラットフォームでサポートされていても、パーミッションの設定の誤りや物理メモリの不足などが原因で、lock shared memory パラメータを設定できないことがあります。共有メモリのロックについては、使用しているプラットフォームに対応するオペレーティング・システムのガイドを参照してください。

lock spinlock ratio

要約	
デフォルト値	85
値の範囲	1 ～ 2147483647
ステータス	静的
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	ロック・マネージャ、メモリ使用

Adaptive Server は、設定可能な数のハッシュ・バケットを持つ内部ハッシュ・テーブルを使用してロックの取得と解放を管理します。対称型マルチプロセッシング・システムでは、このハッシュ・テーブルは 1 つまたは複数のスピンロックを使用して、異なるエンジンで実行しているプロセス間のアクセスを直列化できます。ハッシュ・バケットの数を設定するには、lock hashtable size を使用します。

Adaptive Server が複数のエンジンを実行する場合は、lock spinlock ratio (スピンロック率) によって、1 つのスピンロックで保護されるロック・ハッシュ・バケットの数が決まります。lock hashtable size の値を大きくすると、スピンロックの数は増えますが、1 つのスピンロックで保護されるハッシュ・バケットの数は変わりません。

Adaptive Server の lock spinlock ratio のデフォルト値は 85 です。lock hashtable size がデフォルト値の 2048 に設定されている場合、デフォルトのスピンロック率から計算されるロック・ハッシュ・テーブルに対するスピンロックの数は 26 となります。『システム管理ガイド 第 2 巻』の「第 5 章 マルチプロセッサ・サーバの管理」を参照してください。

sp_sysmon は、ロック・ハッシュ・テーブルのハッシュ・チェーンの平均長についてレポートします。『パフォーマンス＆チューニング・シリーズ：sp_sysmon による Adaptive Server の監視』を参照してください。

lock table spinlock ratio

要約	
デフォルト値	20
値の範囲	1 ～ 2147483647
ステータス	静的
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	ロック・マネージャ

Adaptive Server で複数のエンジンを実行する場合に、table lock spinlock ratio を使用して、1 つのスピンロックによって保護される内部テーブル・ロックのハッシュ・テーブルのロー数を設定します。

Adaptive Server は、101 のローを持つハッシュ・テーブル (ハッシュ・パケット) を使用して、テーブル・ロックの取得と解放を管理します。このテーブルは、1 つまたは複数のスピンロックを使用して、異なるエンジンで実行しているプロセス間のアクセスを直列化できます。

Adaptive Server の lock table spinlock ratio のデフォルト値は 20 で、このときテーブル・ロック・ハッシュ・テーブルに対して 6 個のスピンロックが定義されます。最初の 5 個のスピンロックはそれぞれ 20 ローを保護し、6 番目のスピンロックは最後のローを保護します。lock table spinlock ratio に 101 以上の値を指定すると、Adaptive Server はテーブル全体に対してスピンロックを 1 つだけ使用します。

lock wait period

要約	
デフォルト値	2147483647
値の範囲	0 ～ 2147483647
ステータス	動的
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	ロック・マネージャ

lock wait period は、テーブル、データ・ページ、データ・ローに対するロックを取得するまでのタスクの待機時間 (秒) を制限します。指定した時間内にロックが取得されなかった場合、Adaptive Server はエラー・メッセージ 12205 をユーザに返し、トランザクションをロールバックします。

set コマンドの lock wait オプションは、タスクがロックを待機する秒数をセッション・レベルで設定します。これを設定すると、そのセッションに対するサーバ・レベルの設定が無効になります。

lock wait period は、セッション・レベルの設定 set lock wait nnn と併用され、ユーザ定義テーブルにのみ有効です。これらの設定は、システム・テーブルには影響しません。

デフォルトでは、すべてのプロセスはロックを取得するまで無制限に待機します。デフォルト値に戻すには、値を 2147483647 にリセットするか、または次のように入力します。

```
sp_configure "lock wait period", 0, "default"
```

log audit logon failure

要約	
デフォルト値	0 (オフ)
値の範囲	0 (オフ)、1 (オン)
ステータス	動的
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	エラー・ログ

log audit logon failure は、イベント・ロギングが有効になっている場合に、Adaptive Server へのログインの失敗を Adaptive Server のエラー・ログ (Windows サーバの場合は Windows イベント・ログにも) に記録するかどうかを指定します。

log audit logon success

要約	
デフォルト値	0 (オフ)
値の範囲	0 (オフ)、1 (オン)
ステータス	動的
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	エラー・ログ

log audit logon success は、イベント・ロギングが有効になっている場合に、Adaptive Server へのログインの成功を Adaptive Server のエラー・ログ (Windows サーバの場合は Windows イベント・ログにも) に記録するかどうかを指定します。

max async i/os per engine

要約	
デフォルト値	プラットフォームに依存する
値の範囲	1 〜プラットフォーム依存値
ステータス	静的
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	O/S リソース

max async i/os per engine パラメータは、同時に存在できる処理待ちの非同期ディスク I/O 要求の、エンジン当たりの最大数を指定します。

Linux プラットフォームの場合

Linux プラットフォームの場合、max async i/os per engine は、マシン起動時、Linux OS から各エンジンが予約している非同期 I/O の数を制御します。デフォルト値より大きい数値を使用することで、使用しているシステムに利点をもたらすことができる可能性があります。

sp_sysmon は、max async i/os per engine の調整に役立ちます。sp_sysmon の `disk i/o section` には、サンプル期間中の各エンジンの未処理 I/O の最大数情報と、エンジンやオペレーティング・システムの制限によって処理が遅延した I/O の数に関する情報が含まれています。通常、I/O がエンジン制限で遅延した場合は、max async i/os per engine の値を大きくする必要があります。

Adaptive Server がデバイスに対して非同期 I/O を実行できるかどうかは、このデバイスがカーネル非同期 I/O (KAIO: kernel asynchronous I/O) をサポートしているかどうかによって決まります。Linux カーネルでは、ファイル・システム・レベルで KAIO サポートを実装する必要があります。ext3、xfs、jfs、ロー・デバイスなど、主要なほとんどのファイル・システムで KAIO がサポートされています。Adaptive Server は、KAIO をサポートしていないデバイスに対して非同期 I/O を実行できません。その場合、デバイスを標準の同期 I/O に戻してすべての読み書きを処理します。Adaptive Server は、デバイスが同期 I/O に切り替わったことを示す次のようなメッセージをエラー・ログに記録します。

```
00:00000:00001:2006/12/15 11:47:17.98 kernel Virtual device
'/dev/shm/tempdb.dat' does not support kernel asynchronous i/o.
will be used for this device.
```

max async i/os per server

要約	
デフォルト値	プラットフォームに依存する
値の範囲	1 ～プラットフォーム依存値
ステータス	静的
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	O/S リソース

max async i/os per server は、同時に存在できる処理待ちの非同期ディスク I/O 要求の、Adaptive Server 当たりの最大数を指定します。この制限には、Adaptive Server 当たりのオンライン・エンジンの数は影響しません。max async i/os per engine は、エンジン当たりの未処理 I/O の数を制限します。

ほとんどのオペレーティング・システムは、一度に処理できる非同期ディスク I/O の数を制限しています。オペレーティング・システムのプロセス当たりの数を制限するか、システム当たりの数を制限するか、あるいはその両方を制限するかは、オペレーティング・システムによって異なります。アプリケーションがこれらの制限を超えると、オペレーティング・システムはエラー・メッセージを表示します。オペレーティング・システムの呼び出しは比較的コストがかかるので、オペレーティング・システムによって拒否されるような非同期 I/O を Adaptive Server が実行しようとするのは効率的ではありません。

これを避けるために、Adaptive Server はエンジンとサーバのそれぞれについて、処理待ちの非同期 I/O の数を常に把握しています。あるエンジンが発行した非同期 I/O によって max async i/os per engine と max async i/os per server のどちらかの制限を超えることがわかった場合は、処理待ちの I/O の処理が完了して制限を下回るまで、その I/O を遅延させます。

たとえば、システム当たりの非同期 I/O 数が 200、プロセス当たりの非同期 I/O 数が 75 というオペレーティング・システムの制限があり、Adaptive Server に 3 つのオンライン・エンジンがあるとします。そして、現在の全エンジンの保留中非同期 I/O 数の合計は 200 で、その内訳は次の表のとおりであるとします。

エンジン	保留中の I/O 数	結果
0	60	エンジン 0 は、サーバ当たりの総数がオペレーティング・システムの「システム当たりの」制限値を下回るまで、以後の非同期 I/O を遅延させる。制限値を下回ると、非同期 I/O の発行を再開する。
1	75	エンジン 1 は、エンジン当たりの総数がオペレーティング・システムの「プロセス当たりの」制限値を下回るまで、以後の非同期 I/O を遅延させる。制限値を下回ると、非同期 I/O の発行を再開する。
2	65	エンジン 2 は、サーバ当たりの総数がオペレーティング・システムの「システム当たりの」制限値を下回るまで、以後の非同期 I/O を遅延させる。制限値を下回ると、非同期 I/O の発行を再開する。

非同期 I/O と同期 I/O の両方とも、すべての I/O はディスク I/O 構造体を必要とするので、処理待ちディスク I/O の総数は disk i/o structures の値によって制限されます。Adaptive Server の効率の点では、I/O 要求の数が max i/os per server を超えたことが理由で I/O を遅延させるよりも、ディスク I/O 構造体を取得できないことが理由で遅延させる方が、わずかに勝っています。max async i/os per server は disk i/o structures と同じ値に設定してください。“disk i/o structures” (109 ページ) を参照してください。

非同期 I/O に関するオペレーティング・システムでの制限が調整可能な場合は、Adaptive Server が動作できるように十分大きな値を設定してください。必要なだけ大きい値に設定することによる不利益はありません。

サーバ当たりの制限、またはエンジン当たりの制限によって I/O の遅延が発生しているかどうかを確認するには、sp_sysmon を使用してください。sp_sysmon の結果から、処理待ち要求に関するエンジン当たりまたはサーバ当たりの制限を超えていることがわかった場合は、対応するパラメータの値を大きくします。『パフォーマンス＆チューニング・シリーズ：sp_sysmon による Adaptive Server の監視』を参照してください。

max cis remote connections

要約	
デフォルト値	0
値の範囲	0 ～ 2147483647
ステータス	動的
表示レベル	基本
必要な役割	システム管理者
設定グループ	コンポーネント統合サービス

max cis remote connections は、コンポーネント統合サービスによって確立できる、リモート・サーバへの Client-Library 接続の最大同時接続数を指定します。

デフォルトでは、コンポーネント統合サービスによって確立できるリモート・サーバへの同時接続数は、ユーザ当たり最大 4 つです。ユーザの最大数を 25 に設定している場合は、コンポーネント統合サービスによって最大 100 の Client-Library 接続を同時に確立できます。

この値がインストール環境のニーズを満たしていない場合は、サーバが一度に確立できる Client-Library 接続の正確な数を指定することにより、この設定を上書きできます。

max concurrently recovered db

要約	
デフォルト値	0
有効な値	1 ～ number of engines at start-up から 1 を引いた値
ステータス	動的
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	バックアップとリカバリ

max concurrently recovered db は、並列度を決定します。最小値は 1 であり、この場合は逐次リカバリが行われます。デフォルト値の 0 を指定して、セルフチューニング・アプローチを使用させることもできます。最大値は、number of engines at start-up から 1 を引いた値です。max concurrently recovered db の上限値には、number of open databases 設定パラメータの値も影響します。

max memory

要約	
デフォルト値	プラットフォーム依存
値の範囲	プラットフォーム固有の最小値 ～ 2147483647
ステータス	動的
表示レベル	基本
必要な役割	システム管理者
設定グループ	メモリ使用、物理メモリ

max memory は Adaptive Server が割り付ける物理メモリの総量の最大値を指定します。max memory は、Adaptive Server の現在の設定で使用される論理メモリの総量よりも大きくなければなりません。

コンピュータで使用可能なメモリの最大量を使用するように Adaptive Server を設定しても、パフォーマンスが低下することはありません。ただし、そのシステムで Adaptive Server 以外に必要となるメモリについても検討してください。このようにしないと、Adaptive Server の起動時に必要なメモリを確保できないことがあります。

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

Adaptive Server が起動しない場合

allocate max shared memory が 1 に設定されている場合は、Adaptive Server は max memory で指定された量のメモリを確保しなければなりません。このメモリ量が確保できなければ、Adaptive Server は起動しません。この場合は、サーバの設定ファイルを編集して max memory の値を変更し、Adaptive Server に必要なメモリの量を減らしてください。max memory に指定したメモリ量がすべて確保できなくても起動できるように、allocate max shared memory の値を 0 に変更することもできます。

また、大量のメモリを必要とする他の設定パラメータの値も減らす必要がある場合もあります。その後で Adaptive Server を再起動すると、新しく指定した量のメモリが使用されます。他の設定パラメータの合計値が max memory の値を超えることが理由で Adaptive Server が起動できない場合は、メモリを使用する設定パラメータについて、『システム管理ガイド 第 2 巻』の「第 3 章 メモリの設定」を参照してください。

max native threads per engine

要約	
デフォルト値	50
最大値	50 ～ 1000
ステータス	動的
表示レベル	中間
必要な役割	システム管理者
設定グループ	ユーザ環境

max native threads per engine は、サーバが 1 つのエンジンについて生成するネイティブ・スレッドの最大数を定義します。ネイティブ・スレッドの最大数に達すると、ネイティブ・スレッドを必要とする Adaptive Server セッションは他のセッションがネイティブ・スレッドを解放するまでスリープします。

max nesting level

Adaptive Server 15.0.3 とそれ以降では、最大ネスト・レベルが 100 に引き上げられており、デフォルト値は 50 です。

要約	
デフォルト値	50
値の範囲	16 ～ 100
ステータス	静的
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	ユーザ環境

maximum nesting level は、ストアド・プロシージャとトリガの最大ネスト・レベルを設定します。ネスト・レベルを 1 つ上げるごとに、追加メモリが 160 バイト必要になります。たとえば、ネスト・レベルを 16 から 26 に上げると、追加で 1600 バイトのメモリが必要になります。

max network packet size

要約	
デフォルト値	512
値の範囲	512 ～ 65024
ステータス	静的
表示レベル	中間
必要な役割	システム管理者
設定グループ	ネットワーク通信

max network packet size は、Adaptive Server と通信するクライアントが要求できる最大ネットワーク・パケット・サイズを指定します。

アプリケーションでネットワークを介して大量のデータを送受信する場合は、大きなパケット・サイズを使用すると、アプリケーションのパフォーマンスを大幅に改善できます。例としては、大量のバルク・コピー操作と、大きな text 値、unitext 値、image 値を読み書きするアプリケーションの 2 つがあります。

通常は、次のようにします。

- 短いクエリを実行するユーザについては、default network packet size の値を小さくする。
- 大量のデータを送受信するユーザについては、大きなパケット・サイズを要求できるように、max network packet size を十分に大きく設定する。

max network packet size は、default network packet size と同じかそれよりも大きくなければなりません。512 バイトの整数倍以外の値を指定した場合は、倍数になるように切り捨てられます。

クライアント・アプリケーションからより大きなネットワーク・パケット・サイズが明示的に要求された場合に、そのサイズのパケットをクライアントが受信できるようにするには、additional network memory も設定する必要があります。“additional network memory” (78 ページ) を参照してください。

Open Client Server は、64K を超える大きさのネットワーク・パケットは受信できません。

bcp と isql プログラムで大きなパケット・サイズを使用する方法については、『ユーティリティ・ガイド』の該当するプログラムの項を参照してください。Open Client Client-Library のマニュアルには、可変パケット・サイズの使用法の説明があります。

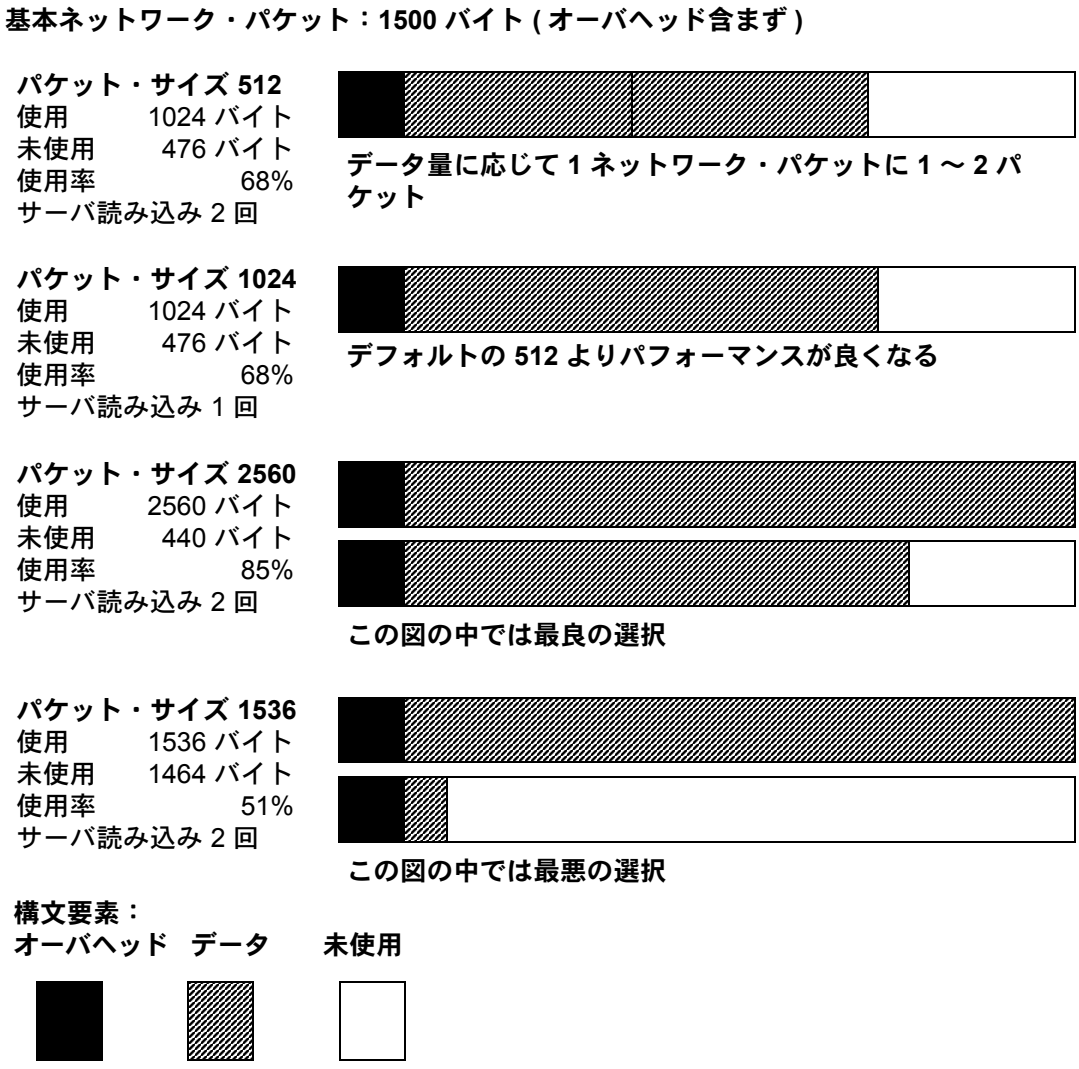
パケット・サイズの選択

最良のパフォーマンスを得るには、ネットワーク上の基本パケット・サイズに対して効率的なサーバ・パケット・サイズを選択してください。目標は次の 2 つです。

- ネットワークに対するサーバの読み込みと書き込みの数を減らす
- ネットワーク・パケット内の未使用領域を減らして、ネットワーク・スループットを向上させる

たとえば、ネットワーク・パケット・サイズが 1500 バイトならば、Adaptive Server のパケット・サイズを 1024 (512 * 2) に設定すると、1536 (512 * 3) 図 5-3 に設定した場合よりもパフォーマンスが向上する可能性が高くなります。

図 *5-3:* パケット・サイズを決定する要因



ネットワーク上の基本パケットの使用可能なデータ領域を計算した後で、ベンチマーク・テストを行い、最適なサイズを決定します。

max network packet size の変更が、ネットワーク I/O 管理とタスク切り替えにどのように影響しているかを確認するには、sp_sysmon を使用してください。たとえば、max network packet size を増やした後で、sp_sysmon の出力をチェックすることにより、bcp で大きいバッチを処理するときにこの設定がどのように影響するかを確認します。『パフォーマンス＆チューニング・シリーズ：sp_sysmon による Adaptive Server の監視』を参照してください。

max number network listeners

要約	
デフォルト値	5
値の範囲	0 ～ 2147483647
ステータス	静的
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	メモリ使用、ネットワーク通信

max number network listeners は、Adaptive Server で同時に使用できるネットワーク・リスナの最大数を指定します。

各マスタ・ポートは、1 つのネットワーク・リスナを持ちます。一般に、Adaptive Server が複数のネットワーク・タイプで通信する必要がある場合を除いて、複数のマスタ・ポートを持つ必要はありません。プラットフォームによっては、ソケットと TLI (トランスポート・レイヤ・インタフェース) の両方のネットワーク・インタフェースをサポートするものもあります。サポートされるネットワーク・タイプの詳細については、使用しているプラットフォームの『設定ガイド』を参照してください。

max online engines

要約	
デフォルト値	1
値の範囲	1 ～ 128
ステータス	静的
表示レベル	中間
必要な役割	システム管理者
設定グループ	メモリ使用、プロセッサ

max online engines の役割は、対称型マルチプロセッサ (SMP) 環境において同時にオンラインになるエンジンの最大数を設定することです。起動時に使用可能な CPU の数は考慮されません。したがって、ユーザは後から CPU を増設することができます。

max engines online パラメータは、SMP 環境下で、同時にオンラインにできる Adaptive Server エンジンの最大数を指定します。このパラメータを各自の SMP 環境に適合するように設定する方法の詳細については、『システム管理ガイド第 2 巻』の「第 5 章 マルチプロセッサ・サーバの管理」を参照してください。

Adaptive Server の起動時は、1 つのエンジンで、すべてのデータベースのリカバリを含む初期化が行われます。初期化での最後のタスクは追加サーバ・エンジンを割り付けることです。それぞれのエンジンは、共有メモリ内の共通データ構造体にアクセスします。

max engines online パラメータを調整する場合は、次の点に注意してください。

- 実装されている CPU の数よりオンライン・エンジン数を多くしないでください。
- Adaptive Server 以外のアプリケーションを含めたシステム全体の負荷によっては、Adaptive Server 以外のプロセスの実行用に一部の CPU を残しておくことで最適なスループットを得られることがあります。
- CPU 使用率の低い多数のエンジンを実行するより、CPU 使用率の高い少数のエンジンを実行する方が、より良いスループットを得ることができます。
- スケーラビリティはアプリケーションによって異なります。アプリケーションについて広範囲なベンチマークを実行して、オンライン・エンジンの最良の設定を決定してください。
- sp_engine を使用して、エンジンのオフラインとオンラインを切り替えることができます。エンジン 0 を除くすべてのエンジンをオフラインにできます。

『パフォーマンス&チューニング・シリーズ：基本』の「第 3 章 エンジンと CPU の使用方法」を参照してください。

max online Q engines

要約	
デフォルト値	0
値の範囲	0 ～ 127
ステータス	静的
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	

max online Q engines は、MQ に必要です。オンラインで実行できる Q エンジンの最大数を指定します。max online Q engines の数値に対応するには、max online engines を増やす必要がある場合があります。

max parallel degree

要約	
デフォルト値	1
値の範囲	1 ～ 255
ステータス	動的
表示レベル	基本
必要な役割	システム管理者
設定グループ	クエリ・チューニング

max parallel degree は、クエリ当たりの使用可能なワーカー・プロセス数の、サーバワイドの最大値を指定します。これを「最大並列度」と呼びます。

max parallel degree が小さすぎると、クエリのパフォーマンスはそれほど向上しないこともあります。また、max parallel degree が大きすぎると、サーバによってコンパイルされたプランに必要なプロセス数が実行時に実際に使用できる数を超えてしまうことや、システムが飽和状態になってスループットが低下してしまうことがあります。並列パーティション・スキャンを有効にするには、クエリを行うテーブル内のパーティション数以上となるようにこのパラメータの値を設定します。

このパラメータの値は、現在の number of worker processes の値以下にしてください。

max parallel degree を 1 に設定すると、次のようになります。

- Adaptive Server はすべてのテーブルまたはインデックスを逐次スキャンする
- Adaptive Server は逐次クエリ実行を強制し、オプティマイザはこの設定が無効にされている場合よりも並列度の高いプランを選択することがある

max parallel degree を変更すると、プロシージャ・キャッシュ内のクエリ・プランはすべて無効になります。新しいプランは、次回のストアド・プロシージャまたはトリガの実行時にコンパイルされます。

『パフォーマンス＆チューニング・シリーズ：クエリ処理と抽象プラン』の「第 9 章 並列ソート」を参照してください。

max pci slots

要約	
デフォルト値	0
値の範囲	0 ～ 30
ステータス	静的
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	ユーザ環境

Adaptive Server で許容される PCI スロットの最大数を設定します。以下の値のいずれかです。

- 0、1 － PCA を 1 つ備えたデフォルト・ブリッジ

  注意 JVM サポートに必要なのは 1 つのスロットです。スロット数を増やさないでください。

- 2 ～ 30 － 今後のリリース用に割り付け

PCI スロットの詳細については、『Adaptive Server Enterprise における Java』を参照してください。

max query parallel degree

要約	
デフォルト値	1
値の範囲	1 ～ 255
ステータス	静的
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	クエリ・チューニング

Adaptive Server が互換モードのときに使用されます。所定のクエリに使用するワーカー・プロセスの数を定義します。このパラメータが関係するのは、並列処理をグローバルに有効にしない場合だけです。number of worker process の値は、max query parallel degree の値より小さくできません。

『パフォーマンス&チューニング・シリーズ：クエリ処理と抽象プラン』の「第 5 章 並列クエリ処理」を参照してください。

互換モードの詳細については、『マイグレーション技術ガイド』を参照してください。

max repartition degree

要約	
デフォルト値	1
値の範囲	1 ～ max parallel degree の値
ステータス	動的
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	クエリ・チューニング

max repartition degree は、Adaptive Server で必要な動的再分割の数を設定します。これにより、Adaptive Server は、水平並列処理を実行できます。ただし、分割の数が多すぎると、リソースを要求するワーカー・プロセスが大量に発生し、結果的にパフォーマンスが低下します。max repartition degree の値は、これらのリソースのために作成されるパーティションの最大数を決定します。すべてのテーブルとインデックスが非分割である場合は、データの再分割の結果として作成されるパーティションの数として max repartition degree の値が使用されます。

max resource granularity

要約	
デフォルト値	10
値の範囲	1 〜 100
ステータス	動的
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	クエリ・チューニング

max resource granularity は、システム・リソースの何パーセントまでをクエリが使用できるかを示す上限値を示します。このパラメータは、実行時には使用されませんが、クエリ・オプティマイザにとっての参考値としてのみ使用されるため、クエリ・プロセッサでクエリを並列実行できなくなることはありません。クエリ・エンジンは、max resource granularity を参考値として使用することで、メモリの集中使用を回避することができます。

max scan parallel degree

要約	
デフォルト値	1
値の範囲	1 〜 255
ステータス	動的
表示レベル	基本
必要な役割	システム管理者
設定グループ	クエリ・チューニング

max scan parallel degree は、ハッシュベースのスキャンの最大並列度をサーバワイドで指定します。これは、次のアクセス・メソッドで使用することがあります。

- 分割テーブルと非分割テーブルの並列インデックス・スキャン
- 非分割テーブルの並列テーブル・スキャン

max scan parallel degree は、テーブルごとまたはインデックスごとに適用されます。つまり、max scan parallel degree が 3 に設定されており、ジョイン・クエリ内の 1 つのテーブルがハッシュ・テーブルを使用してスキャンされ、別のテーブルがハッシュベースのインデックス・スキャンによってアクセスされる場合は、このクエリで 9 個のワーカー・プロセスを使用できます (max scan parallel degree が 9 以上に設定されている場合)。

オプティマイザは、非パーティションベースの並列スキャン処理で使用するプロセス数を選択するときに、max scan parallel degree をガイドラインとして使用します。これは並列ソートには適用されません。分割によってデータが複数のデバイスに分散することはないので、複数の並列プロセスがスキャン中に同じデバイスにアクセスすることがあります。そのため、ディスクの競合とヘッドの移動が増え、その結果パフォーマンスが低下する場合があります。複数のディスク・アクセスによる問題を防ぐには、max scan parallel degree を使用して、テーブルに同時にアクセスできるプロセスの最大数を減らします。

この値が小さすぎると、クエリのパフォーマンスはそれほど向上しません。この値が大きすぎると、サーバによってコンパイルされたプランが使用するプロセス数が多すぎて、ディスク・アクセスの効率が低下してしまうこともあります。一般的には、このパラメータには 2 ～ 3 以下の値を設定します。これは、1 つの物理デバイスの I/O を完全に利用するには、ワーカー・プロセスは 2 ～ 3 個で十分だからです。

max scan parallel degree の値は、max parallel degree の現在値以下に設定してください。max parallel degree よりも大きな値を指定すると、Adaptive Server のエラーが返されます。

max scan parallel degree が 1 に設定されているときは、Adaptive Server はハッシュベースのスキャンは実行しません。

max scan parallel degree を変更すると、プロシージャ・キャッシュ内のクエリ・プランはすべて無効になります。新しいプランは、次回のストアド・プロシージャまたはトリガの実行時にコンパイルされます。

max SQL text monitored

要約	
デフォルト値	0
値の範囲	0 ～ 2147483647
ステータス	静的
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	メモリ使用、モニタリング

max SQL text monitored は、Adaptive Server Monitor と共有しているメモリに SQL テキストを保存するためにユーザ接続ごとに割り付けられるメモリ量を指定します。

バッチ文用に十分なメモリを割り付けないと、表示したいテキストがトランケートされてしまうことがあります。Sybase では、1 ユーザ接続当たりのメモリ初期値を 1,024 バイトにすることをおすすめします。

共有メモリから SQL テキスト用に割り付けられる合計メモリ量は、max SQL text monitored にユーザ接続数の現在の設定値を掛けた値です。

「SQL バッチ・テキストを保存するための Adaptive Server の設定」(355 ページ) を参照してください。

max transfer history

要約	
デフォルト値	10
値の範囲	1 ～ 255
ステータス	動的
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	Adaptive Server 管理

max transfer history は、Adaptive Server が各データベースの spt_TableTransfer テーブルに保持する転送履歴の数を制御します。追跡される各テーブルについて、spt_TableTransfer は次のように保持します。

(*N* 個の成功したエントリ) + (*N* 個の失敗したエントリ)

ただし、*N* は max transfer history の値です。

このパラメータの値を下げても、spt_TableTransfer からエントリが自動的に削除されるわけではありません。所定の転送テーブルのエントリが削除されるのは、次回そのテーブルの転送を開始するときです。転送に成功すると、テーブルの成功転送エントリがクリアされます。転送に失敗すると、失敗した転送エントリがクリアされます。

たとえば、ある 1 つのテーブルの spt_configure に 12 個の成功履歴エントリと 9 個の失敗履歴エントリがあり、max transfer history が 5 に設定された場合、そのテーブルで次回の転送に成功すると、spt_configure での成功エントリ数が 5 になりますが、前回の失敗エントリ数 9 はそのまま変わりません。

maximum dump conditions

要約	
デフォルト値	10
値の範囲	10 ～ 100
ステータス	静的
表示レベル	中間
必要な役割	システム管理者
設定グループ	グループ診断

maximum dump conditions は、Adaptive Server が共有メモリ内にデータのダンプを生成する条件の最大数を設定します。

注意 このパラメータは、Sybase 製品の保守契約を結んでいるサポート・センタだけが使用します。このパラメータは、Sybase 製品の保守契約を結んでいるサポート・センタから指示がないかぎり、変更しないでください。

max buffers per lava operator

要約	
デフォルト値	2048
値の範囲	500 ～ 65535
ステータス	動的
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	クエリ・チューニング

max buffers per lava operator は、lava 演算子が使用するバッファの数の上限値を設定します。この演算子は、ソートまたはハッシュを実行します (これらは、処理の観点からは「高コスト」です)。lava 演算子はセッションの tempdb データ・キャッシュ・プールのバッファを、ローを処理するための作業領域として使用します。

lava 演算子は入力ストリームを再帰処理することがよくあります。ソートでは、残りの実行すべてをマージするのに十分なバッファが確保されるまで、それ以降のマージ・パスが必要とされます。ハッシュでは、残りのデータすべてが 1 つのメモリ内ハッシュ・テーブルにちょうど収まるまで、こぼれ落ちたセットのハッシュ・テーブルを作成するため、それ以降のパスが必要とされます。max buffers per lava operator の値を増やした場合、一部のクエリでは必要な I/O が減少します。これは特に、HASH DISTINCT、HASH VECTOR AGGREGATE、HASH UNION の各演算子を使用するクエリに当てはまります。

同時ユーザが多くいるサーバーで max buffers per lava operator のデフォルト値を増やすときには、注意を払ってください。Adaptive Server が、コストの高い演算子だけにバッファを割り付けるので、ユーザのテーブルや他のセッションのワーク・テーブルのキャッシュに使用できるバッファの数が減少してしまう可能性があるからです。tempdb のデータ・キャッシュの効果を分析するには、sp_sysmon を使用します。

max buffers per lava operator は、max resource granularity とともにバッファの使用数を制限します。制限値は、次の項目の最低値に設定されます。

- max buffers per lava operator の値、または
- (max resource granularity) X (tempdb のページサイズ・プールにおけるデータ・バッファ数)

ソート・バッファに割り付けられるメモリ量の設定については、「number of sort buffers」(195 ページ) を参照してください。

maximum failed logins

要約	
デフォルト値	0
値の範囲	-1 ～ 32767
ステータス	動的
表示レベル	10
必要な役割	システム・セキュリティ担当者
設定グループ	セキュリティ関連

maximum failed logins は、ログインや役割に対するログイン試行の最大回数をサーバワイドで設定するときに使用します。

値 "-1" は、認証に失敗するたびに syslogins カラムの logincount のログイン失敗回数は更新されても、アカウントはロックされないことを指定します。一方、値 0 (ゼロ) を使用すると、認証失敗のたびにカラム数が増えることはなく、認証失敗によってアカウントがロックされることもありません。

sp_modifylogin を使用して特定の 1 つの役割に対する最大失敗ログイン回数を変更する方法の詳細については、『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。alter role を使用して最大失敗ログイン回数を変更する方法の詳細については、『リファレンス・マニュアル：コマンド』を参照してください。

maximum job output

要約	
デフォルト値	32768
値の範囲	0 ～ 2147483647
ステータス	動的
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	SQL Server 管理

maximum job output は、1 つのジョブが生成できる出力の最大サイズをバイト単位で設定します。maximum job output で指定された値を上回る出力が生成された場合、超過した分のデータは破棄されます。

memory alignment boundary

要約	
デフォルト値	論理ページ・サイズ
値の範囲	2048[a] ～ 16384 a. 最小値はサーバの論理ページ・サイズによって決まる
ステータス	静的
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	キャッシュ・マネージャ

memory alignment boundary は、データ・キャッシュを揃えるメモリ・アドレス境界を決定します。

構造体が特定のメモリ・アドレス境界に揃えられていれば、I/O のパフォーマンスが向上するマシンもあります。この整列を維持するためには、memory alignment boundary の値が論理ページ・サイズから 2048K までの範囲にある 2 の累乗でなければなりません。

注意　memory alignment boundary がサポートされるのは、特定のハードウェア・プラットフォームだけです。このパラメータは、Sybase 製品の保守契約を結んでいるサポート・センタから指示がないかぎり、変更しないでください。

memory per worker process

要約	
デフォルト値	1024
値の範囲	1024 ～ 2147483647
ステータス	動的
表示レベル	基本
必要な役割	システム管理者
設定グループ	メモリ使用

memory per worker process は、ワーカー・プロセスが使用するメモリ量をバイト単位で指定します。各ワーカー・プロセスは、クエリの処理中にメッセージ用のメモリを要求します。このメモリは、共有メモリ・プールから割り付けられます。このプールのサイズは、memory per worker process と number of worker processes を乗じた値です。並列処理の使用状況に応じて 2 倍から 4 倍の値に再設定することを必要とする場合があります。dbcc checkstorage を使用するときに、number of worker processes の値が 1 に設定されている場合は、memory per worker process を 1,792 バイトに増やす必要があります。

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

messaging memory

要約	
デフォルト値	400
値の範囲	60 ～ 2147483647
ステータス	動的
表示レベル	中間
必要な役割	システム管理者
設定グループ	メモリ使用、物理メモリ

Sybase メッセージングに使用できるメモリ量を設定します。

metrics elap max

要約	
デフォルト値	0
値の範囲	0 ～ 2147483647
ステータス	動的
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	クエリ・チューニング

metrics elap max は、QP 測定基準の最長経過時間とスレッショルドを設定します。

metrics exec max

要約	
デフォルト値	0
値の範囲	0 ～ 2147483647
ステータス	動的
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	クエリ・チューニング

metrics exec max は、QP 測定基準の最長実行時間とスレッショルドを設定します。

metrics lio max

要約	
デフォルト値	0
値の範囲	0 ～ 2147483647
ステータス	動的
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	クエリ・チューニング

metrics lio max は、QP 測定基準の最大論理 I/O とスレッショルドを設定します。

metrics pio max

要約	
デフォルト値	0
値の範囲	0 ～ 2147483647
ステータス	動的
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	

metrics pio max は、QP 測定基準の最大物理 I/O とスレッショルドを設定します。

min pages for parallel scan

要約	
デフォルト値	200
値の範囲	20 ～ 2147483647
ステータス	動的
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	クエリ・チューニング

min pages for parallel scan は、Adaptive Server が並列にアクセスできるテーブルとインデックスの数を制御します。テーブルのページ数が設定値より低い場合、テーブルは逐次アクセスされます。min pages for parallel scan はページ・サイズを考慮しません。Adaptive Server がインデックスとテーブルにアクセスするとき、適切であればデータを再分割します。そして、適切であればスキャン数を超える並列処理を使用します。

minimum password length

要約	
デフォルト値	6
値の範囲	0 ～ 30
ステータス	動的
表示レベル	10
必要な役割	システム・セキュリティ担当者
設定グループ	セキュリティ関連

minimum password length を使用すると、パスワード値の長さを、サーバワイドで、あるいはログインごとまたは役割ごとにカスタマイズできます。ログインごとまたは役割ごとの minimum password length 値は、サーバワイドの設定よりも優先します。minimum password length の設定は、値を設定した後に作成するパスワードにのみ影響します。既存のパスワードの長さは変更されません。

minimum password length を使用して指定するサーバワイドの minimum password length の値は、ログインと役割の両方に適用されます。たとえば、すべてのログインと役割に対する minimum password length を 4 文字に設定するには、次のように入力します。

```
sp_configure "minimum password length", 4
```

ログインの作成時にそのログインの minimum password length を設定するには、sp_addlogin を使用します。たとえば、パスワードが “Djdiek3” である新しいログイン “joe” を作成し、“joe” の minimum password length を 4 に設定するには、次のように入力します。

```
sp_addlogin joe, "Djdiek3", minimum password length=4
```

役割の作成時にその役割の minimum password length を設定するには、create role を使用します。パスワードが “temp244” である新しい役割 “intern_role” を作成し、“intern_role” の minimum password length を 0 に設定するには、次のように入力します。

```
create role intern_role with passwd "temp244", minimum password length 0
```

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

既存のログインに対する minimum password length を設定または変更するには、sp_modifylogin を使用します。sp_modifylogin 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。たとえば、ログイン “joe” に対する minimum password length を 8 文字に設定するには、次のように入力します。

```
sp_modifylogin "joe", @option="mininum password length", @value="8"
```

注意 *value* パラメータのデータ型は character です。したがって、数値には引用符が必要です。

すべてのログインに対する minimum password length のオーバライドを 2 文字に変更するには、次のように入力します。

```
sp_modifylogin "all overrides", "minimum password length", @value="2"
```

すべてのログインに対する minimum password length のオーバライドを削除するには、次のように入力します。

```
sp_modifylogin "all overrides", @option="minimum password length", @value="-1"
```

既存の役割に対する minimum password length を設定または変更するには、alter role を使用します。たとえば、既存の役割 “physician_role” に対する minimum password length を 5 文字に設定するには、次のように入力します。

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

すべての役割の minimum password length を無効にするには、次のように入力します。

```
alter role "all overrides" set minimum password length -1
```

mnc_full_index_filter

要約	
デフォルト値	2
値の範囲	0 ～ 2 • 0 − 無効化 • 1 − 有効化 • 2 − 最適化目標設定に従って設定
ステータス	動的
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	クエリ・チューニング

mnc_full_index_filter は、次の項目がある場合に、サーバ・レベルでの制限探索引数を持たないカバーされていないインデックスを Adaptive Server が考慮しないように指定します。

- インデックス内のカラム
- ヒストグラムがない述語

インテリジェントなインデックス・スキャンがあるデータオンリーロック (DOL: Data-Only-Locked) テーブルに対して mnc_full_index_filter を使用できます。これは、インテリジェントなインデックス・スキャンが探索引数を作成する場合でも当てはまります。

mnc_full_index_filter の値を変更しても、Adaptive Server で使用するメモリの量は増加しません。

mnc_full_index_filter は、どの特定のオプション目標についても最初は有効にされていません。動作させるには、明示的に有効にする必要があります。

msg confidentiality reqd

要約	
デフォルト値	0 (オフ)
値の範囲	0 (オフ)、1 (オン)
ステータス	動的
表示レベル	中間
必要な役割	システム・セキュリティ担当者
設定グループ	セキュリティ関連

msg confidentiality reqd は、Adaptive Server が送受信するメッセージがすべて暗号化されていなければならないことを指定します。メッセージを暗号化するには、use security services パラメータを 1 にします。

msg integrity reqd

要約	
デフォルト値	0 (オフ)
値の範囲	0 (オフ)、1 (オン)
ステータス	動的
表示レベル	中間
必要な役割	システム・セキュリティ担当者
設定グループ	セキュリティ関連

msg integrity reqd は、すべてのメッセージのデータ整合性が検査されていなければならないことを指定します。メッセージの整合性検査を行うには、use security services を 1 に設定します。msg integrity reqd が 1 に設定されているときは、クライアントから Adaptive Server への接続は、クライアントが特定のセキュリティ・サービスを使用していない場合にかぎって許可されます。許可されないセキュリティ・サービスとして、message integrity、replay detection、origin checks、out-of-seq checks があります。

net password encryption required

要約	
デフォルト値	0
値の範囲	0 ～ 2
ステータス	動的
表示レベル	中間
必要な役割	システム・セキュリティ担当者
設定グループ	セキュリティ関連

net password encryption reqd ログイン認証に RSA 暗号化アルゴリズムまたは Sybase 独自の暗号化アルゴリズムのみを使用するように制限します。表 5-3 に net password encryption reqd の有効な値を示します。

表 *5-3: net password encryption reqd* の値とその説明

値	説明
0	クライアントがネットワーク・ログイン・パスワードに使用する暗号化アルゴリズムを選択できます。暗号化しないオプションもあります。
1	ネットワーク・ログイン・パスワードの暗号化に、クライアントが RSA または Sybase 独自の暗号化アルゴリズムのみを使用するように制限します。これは、以前に接続があったクライアントには Sybase 独自のアルゴリズムを使用した再接続、新しいクライアントにはより強力な RSA アルゴリズムを使用した接続を可能にする、制限が段階的に強化されていく設定です。パスワードの暗号化を使わずに接続しようとするクライアントは接続できません。
2	ネットワーク・ログイン・パスワードの暗号化に、クライアントが RSA 暗号化アルゴリズムのみを使用するように制限します。これは、パスワードの強力な RSA 暗号化を実現します。RSA 暗号化を使わずに接続しようとするクライアントは接続できません。

ネットワーク・パスワードの暗号化が必須であるために接続を拒否された場合、クライアントは次のようなメッセージを受け取ります。

```
Msg 1640, Level 16, State 2:
Adaptive Server requires encryption of the login
password on the network.
```

number of alarms

要約	
デフォルト値	40
値の範囲	40 ～ 2147483647
ステータス	動的
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	メモリ使用、SQL Server 管理

number of alarms パラメータは、Adaptive Server によって割り付けられるアラーム構造体の数を指定します。

Transact-SQL のコマンド waitfor は、文ブロック、ストアド・プロシージャ、またはトランザクションを実行するための、特定の時刻、時間の長さ、またはイベントを定義します。Adaptive Server は、waitfor コマンドを正しく実行するためにアラームを使用します。他の内部処理にもアラームが必要です。

Adaptive Server が必要とするアラーム数が、現在割り付けられている数よりも多い場合は、次のメッセージがエラー・ログに書き込まれます。

```
uasetalarm: no more alarms available
```

各アラーム構造体に必要なメモリのバイト数は小さい値です。number of alarms value の値を非常に大きくする場合は、max memory をそれに見合うように設定してください。

number of aux scan descriptors

要約	
デフォルト値	200
値の範囲	0 ～ 2147483647
ステータス	動的
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	メモリ使用、SQL Server 管理

number of aux scan descriptors は、サーバ上のすべてのユーザが共有するプールで使用できる補助スキャン記述子の数を設定します。

ユーザ接続とワーカー・プロセスにはそれぞれ 48 のスキャン記述子が排他的に割り付けられています。48 のスキャン記述子のうち、16 個がユーザ・テーブル用に予約され、12 個がワーク・テーブル用に予約され、20 個がシステム・テーブル用に予約されます (この 20 個のうち 4 個はロールバック条件用に確保されます)。クエリによって直接的または間接的に参照されるテーブルごとに 1 つの記述子が必要です。ユーザ・テーブルの場合、テーブル参照に含まれるものは次のとおりです。

- クエリの from 句で参照されるすべてのテーブル
- クエリで指定するビューで参照されるすべてのテーブル (ビュー自体はカウントされません)
- サブクエリで参照されるすべてのテーブル
- 参照整合性を検査する必要があるすべてのテーブル (挿入、更新、削除のみに使用されます)
- select...into を使用して作成されたテーブル
- クエリ用に作成されたすべてのワークテーブル

セルフジョイン、複数のビュー、または複数のサブクエリなどで同じテーブルが複数回参照される場合、そのテーブルは参照のたびにカウントされます。クエリに union が含まれる場合、その union クエリの各 select 文は個別のスキャンになります。クエリを並列実行する場合は、コーディネーティング・プロセスとそれぞれのワーカー・プロセスにはテーブル参照ごとにスキャン記述子が必要です。

クエリ・スキャンによって参照されるユーザ・テーブルの数が 16 を超える場合やワーク・テーブルの数が 12 を超える場合は、共有プールからのスキャン記述子が割り付けられます。インデックス・スキャンではなくテーブル・スキャンでデータオンリーロック・テーブルがアクセスされる場合も、データオンリーロック・テーブルごとに 1 つのシステム・テーブル記述子が必要です。クエリでテーブル・スキャンを使用してスキャンされるデータオンリーロック・テーブルの数が 16 を超える場合は、補助スキャン記述子がスキャンされたテーブルに割り付けられます。

スキャンで、割り付けられている数を使い果たして補助スキャン記述子が必要になったときに、共有プール内に使用できる記述子がない場合、Adaptive Server はエラー・メッセージを表示し、ユーザ・トランザクションをロール・バックします。

クエリで追加のスキャン記述子が必要となることがなくても、システムの稼働条件が増大した場合に備えて、number of aux scan descriptors をデフォルト値の設定のままにしておいてもかまいません。システムのユーザが実行するクエリで 16 を超えるテーブルが絶対に使用されないことと、使用するテーブルに参照整合性制約がほどんどまたはまったくないことが確かな場合のみに、この値を 0 に設定します。「スキャン記述子の使用率のモニタリング」(176 ページ) を参照してください。

クエリがより多くのスキャン記述子を必要とする場合は、次のいずれかの方法に従って問題を解決してください。

- クエリを書き換えるか、テンポラリ・テーブルを使用してクエリをステップに分割する。データオンリーロック・テーブルでは、テーブル・スキャンが多い場合にはインデックスを追加する。
- 使用する参照整合性制約が多い場合は、使用するスキャン記述子が少なくなるようにテーブルのスキーマを再設計する。クエリを実行する前に set showplan、noexec on を有効にすることで、クエリがスキャン記述子をいくつ使用するかが確認できる。
- number of aux scan descriptors 設定の値を増やす。

以降の項では、記述子の不足を避けるために sp_monitorconfig を使用して現在の使用数と最大使用数をモニタリングする方法と、必要なスキャン記述子の数を見積もる方法について説明します。

スキャン記述子の使用率のモニタリング

sp_monitorconfig は、未使用 (空き) のスキャン記述子の数、現在使用されている補助スキャン記述子の数、アクティブになっている割合、サーバの最後の起動以降に使用されたスキャン記述子の最大数をレポートします。スキャン記述子の使用状況をモニタリングするには、ピーク時に周期的に実行します。

次に、500 の記述子が設定されているスキャン記述子の使用状況の例を示します。

```
sp_monitorconfig "aux scan descriptors"

Usage Information at date and time: Apr 22 2002  2:49PM.
Name                num_free   num_active pct_act         Max_Used Reused
--------------      --------  ---------  --------        -------- ------
number of aux            260         240  48.00                427  NA
```

補助スキャン記述子のうち 240 だけが使用され、260 が未使用のままです。ただし、Adaptive Server の最後の起動以降の特定の時点におけるスキャン記述子の最大数は 427 で、残り約 20 パーセント分は、使用が増加したり、例外的に大量に使用される期間が発生したりしても対応できます。“Re-used” はスキャン記述子には適用されません。

補助スキャン記述子の見積もりと設定

スキャン記述子の使用数を見積もるには、次の手順に従います。

1 set showplan と set noexec を有効にしてクエリを実行することによって、17 以上のユーザ・テーブルを参照するクエリや、多数の参照整合性制約を持つテーブルを参照するクエリでのテーブル参照の数を調べます。補助スキャン記述子が要求されると、showplan によって必要な数がレポートされます。

```
Auxiliary scan descriptors required: 17
```

レポートされる数には、すべてのワーカー・プロセスに必要なものも含め、クエリに必要な補助スキャン記述子のすべてが含まれています。クエリに参照整合性制約のみが関係する場合は、sp_helpconstraint も使用できます。これは、テーブルごとの参照整合性制約の数の総数を表示します。

2 補助スキャン記述子を使用するクエリごとに、そのクエリを同時に実行するユーザの数を見積もり、その数を乗算します。補助スキャン記述子が 8 個必要なクエリを実行するユーザ数を 10 と想定すると、合計 80 個が一度に必要になります。

3 クエリごとの結果を合計して、必要な補助スキャン記述子の数を計算します。

number of backup connections

要約	
デフォルト値	0
値の範囲	1 ～ 32768
ステータス	動的
表示レベル	基本
必要な役割	システム管理者
設定グループ	ユーザ環境

number of backup connections は、Backup Server がイン・メモリ・データベースをダンプまたはロードするために確立するユーザ接続の最大数を設定します。number of backup connections の値は、アーカイブされたデータベース当たりの最大ストライプ数を制限します。これが必要なのは、dump または load database が実行されるときに Backup Server ではストライプ当たり 1 つのユーザ接続が必要とされ、しかも dump database コマンドの実行にもう 1 つ追加の接続が必要とされるからです。

number of backup connections は制限値にすぎないので、リソースを消費しません。number of backup connections が 0 に設定されると、Backup Server はユーザ接続の最大数を使用できます。

number of ccbs

要約	
デフォルト値	0
値の範囲	0 ～ 100
ステータス	静的
表示レベル	
必要な役割	
設定グループ	診断

今後のために予約済み。

number of checkpoint tasks

要約	
デフォルト値	1
有効な値	1 ～ 8
ステータス	動的
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	バックアップとリカバリ

number of checkpoint tasks は、並列チェックポイントを設定します。チェックポイント・タスクの値は、起動時のエンジン数の値以下にする必要があります。最大値は、設定パラメータ number of engines online at startup と number of open databases の値によって制限され、最大値は 8 です。

デフォルト値は、逐次チェックポイントをデフォルト動作として設定します。

number of devices

要約	
デフォルト値	10
値の範囲	1 ～ 2,147,483,647
ステータス	動的
表示レベル	基本
必要な役割	システム管理者
設定グループ	ディスク I/O、メモリ使用

number of devices は、Adaptive Server が使用できるデータベース・デバイスの数を制御します。これは、データベースまたはトランザクション・ログ・ダンプ用に使用されるデバイスを含みません。

disk init を実行するときに、仮想デバイス番号 (vdevno) を割り当てることもできます。ただし、この値はオプションです。vdevno を割り当てなかった場合には、次に利用可能な仮想デバイス番号が割り当てられます。

仮想デバイス番号を割り当てる場合、各デバイス番号は Adaptive Server が使用するデバイス番号の中でユニークでなければなりません。番号 0 はマスタ・デバイス用に予約されています。値の有効範囲内にある未使用のデバイス番号を入力できます。

現在使われている番号を調べるには、次のように入力します。

```
select vdevno from master..sysdevices
   where status & 2 = 2
```

ここで、“status 2” は物理ディスクを示します。

注意 UNIX プラットフォーム：多数のデバイスを使用する場合は、適切なデバイス数とユーザ接続数を設定ファイルの中に指定した後、Adaptive Server を再起動することをおすすめします。大量のデバイスを sp_configure を使用して動的に設定する処理は失敗する場合があります。

number of dtx participants

要約	
デフォルト値	500
有効な値	100 ～ 2147483647
ステータス	動的
表示レベル	10
必要な役割	システム管理者
設定グループ	DTM 管理、メモリ使用

number of dtx participants は、Adaptive Server トランザクション・コーディネーション・サービスが同時に送信とコーディネートができるリモート・トランザクションの総数を設定します。DTX パティシパントは、コーディネーション・サービスがリモート・トランザクション分岐の管理に使用する内部メモリ構造です。トランザクションがリモート・サーバに送信されるとき、コーディネーション・サービスはその分岐を管理するために新しい DTX パティシパントを取得する必要があります。

number of dtx participants の設定値をデフォルト値より小さくすると、サーバが管理できるリモート・トランザクションの数が減少します。使用できる DTX パティシパントがない場合、新しい分散トランザクションは開始できません。新しいリモート・トランザクションを送信するために使用できる DTX パティシパントがない場合は、進行中の分散トランザクションがアボートすることがあります。

number of dtx participants の設定値をデフォルト値より大きくすると、Adaptive Server が処理できるリモート・トランザクション分岐の数が増加しますが、メモリの消費量も増加します。

使用しているシステムに対する DTX パティシパント数の最適化

ピークの時間帯に、sp_monitorconfig を使用して DTX パティシパントの使用状況を調査します。

```
sp_monitorconfig "number of dtx participants"

Usage Information at date and time: Apr 22 2002  2:49PM.
Name             num_free  num_active   pct_act    Max_Used   Reused
--------------   -------- ----------  ---------   --------   ------
number of dtx          80          20      4.00         210     NA
```

num_free の値がゼロまたは非常に小さい場合は、新しい分散トランザクションは DTX パティシパントの不足により、開始できない可能性があります。この場合は、number of dtx participants の値を増やしてください。

Max_used の値が低い場合、未使用の DTX パティシパントが他のサーバ機能によって使用できるメモリを消費している可能性があります。この場合には、number of dtx participants の値を減らしてください。

number of dump threads

要約	
デフォルト値	無効
値の範囲	1 (無効、並列なし) ～ 8 (完全に並列)
ステータス	動的
表示レベル	中間
必要な役割	システム管理者
設定グループ	グループ診断

number of dump threads は、Adaptive Server がメモリ・ダンプを実行するために生成するスレッド数を制御します。ダンプ・スレッド数を適切な値にすると、メモリ・ダンプ中にエンジンが停止する時間を短縮できます。

メモリのスレッド数を決めるときは、次の点を考慮してください。

- マシンのファイル・システム・キャッシュにメモリ・ダンプ全体を格納するのに十分な空きメモリがある場合は値 8 を使用します。
- マシンに十分な空きメモリがあるかどうかわからない場合は、ダンプ・スレッド数の値は、I/O システムの速度、ディスクの速度、コントローラのキャッシュ、ダンプ・ファイルがいくつかのディスクで作成された論理ボリューム・マネージャに存在するかどうかなど、多くの要因によって決まります。
- メモリ・ダンプの実行中にエンジンを停止しない場合は、次に説明するように、並列処理を無効化 (値を 1 に設定) します。

  Adaptive Server がメモリ・ダンプを実行するとき、作成されるファイル数は、割り当てたメモリ・セグメントの合計数と設定したスレッド数を掛けた値になります。Adaptive Server は、別々のスレッドを使用して別々のファイルに書き込みます。ジョブが完了すると、エンジンが再起動され、これらのファイルがターゲット・ダンプ・ファイルにマージされます。このため、共有メモリを並列でダンプするときにかかる時間は逐次処理よりも長くなります。
- メモリのダンプ中にエンジンを停止する場合は、1 以外の値を使用すると、メモリ・ダンプ時のエンジンの停止時間を短縮できます。

number of engines at startup

要約	
デフォルト値	1
値の範囲	1 ～ マシンの CPU 数
ステータス	静的
表示レベル	基本
必要な役割	システム管理者
設定グループ	Java サービス、メモリ使用、プロセッサ

Adaptive Server では、ユーザがエンジン 0 を除くすべてのエンジンをオフラインにできます。

number of engines at startup は、起動時にのみ使用されるパラメータで、オンラインにするエンジンの数を設定します。このパラメータで、オンラインにするエンジン数をユーザが自由に設定できます。ただし、マシンに搭載された CPU の数を超える値や、max online engines の設定値を超える値を number of engines at startup で設定することはできません。起動後にエンジンをオンラインに切り替えることをしない場合は、max online engines と number of engines at startup を同じ値に設定します。number of engines at startup と max online engines の値が異なる場合は、1 つのエンジンにつき約 1.8 MB のメモリが浪費されます。

number of histogram steps

要約	
デフォルト値	20
値の範囲	3 ～ 2147483647
ステータス	動的
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	クエリ・チューニング

number of histogram steps は、ヒストグラム内のステップ数を指定します。

number of index trips

要約	
デフォルト値	0
値の範囲	0 ～ 65535
ステータス	動的
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	キャッシュ・マネージャ

number of index trips は、古くなったインデックス・ページが何回 MRU/LRU (Most Recently Used/Least Recently Used) チェーンを周回してから追い出されるかを指定します。number of index trips パラメータの値を増やすに従って、インデックス・ページがキャッシュ内にとどまる時間が長くなります。

データ・キャッシュは MRU/LRU チェーンとして実装されます。ユーザのスレッドがデータ・ページとインデックス・ページにアクセスすると、これらのページはキャッシュの MRU/LRU チェーンの MRU 側の端に置かれます。トランザクション量の多い環境や一部のベンチマークでは、インデックス・ページがすぐに再び必要になる可能性が高いので、キャッシュ内にとどめておいても良いでしょう。number of index trips の設定値が大きいほどインデックス・ページはキャッシュ内に長くとどまり、設定値が小さければインデックス・ページはすぐにキャッシュから追い出されます。

リラックス LRU ページに対しては、number of index trips を設定する必要はありません。『システム管理ガイド 第 2 巻』の「第 4 章 データ・キャッシュの設定」を参照してください。

注意 インデックスが使用するキャッシュが比較的少量で (特に他のオブジェクトと領域を共有する場合)、トランザクション量が多い場合は、number of index trips をあまり大きな値に設定しないでください。追い出されないページでキャッシュがいっぱいになり、プロセスがキャッシュ領域の空きを待っている間にタイムアウトになってしまう可能性があります。
number of index trips の値を 0 以外に変更する前に、すべてのインデックス、OAM、データ・ページを保管できるだけの十分なキャッシュがアプリケーションにあることを確認してください。number of index trips の値を変更する前に、Sybase 製品の保守契約を結んでいるサポート・センタまで連絡してください。

number of java sockets

要約	
デフォルト値	0
有効な値	0 ～ 32767
ステータス	動的
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	Java サービス、メモリ使用

number of java sockets は、Sybase がサポートする Java VM および java.net クラスを有効にします。

number of large i/o buffers

要約	
デフォルト値	6
有効な値	1 ～ 256
ステータス	動的
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	ディスク I/O、メモリ使用、SQL Server 管理

number of large i/o buffers は、特定の Adaptive Server ユーティリティにおける大容量 I/O の実行用に予約される、アロケーション・ユニット・サイズのバッファの数を設定します。大容量 I/O バッファは主に load database コマンドで使用されます。このコマンドは、指定されたストライプ数に関係なく、1 つのバッファを使用してデータベースをロードします。これらのバッファは load transaction では使用されません。6 個を超える load database コマンドを同時に実行するには、load database コマンドごとに 1 つの大容量 I/O バッファを設定します。

create database と alter database は、データベース・ページをクリアするときに、これらの大容量 I/O バッファを使用します。create database および load database の各インスタンスは、最大 32 個の大容量 I/O バッファを使用できます。

これらのバッファは、ディスク・ミラーリングといくつかの dbcc コマンドによっても使用されます。

注意　Adaptive Server バージョン 12.5.0.3 以降では、大容量 I/O バッファのサイズは 1 エクステント (8 ページ) ではなく 1 アロケーション (256 ページ) です。このため、大容量バッファに対してこれまでより大きなメモリ割り付けが必要です。たとえば、以前のバージョンでディスク・バッファとして 8 ページのメモリが必要だった場合に、今回のバージョンでは 256 ページのメモリが必要となります。

number of locks

要約	
デフォルト値	5000
値の範囲	1000 ～ 2147483647
ステータス	動的
表示レベル	基本
必要な役割	システム管理者
設定グループ	ロック・マネージャ、メモリ使用

number of locks は、Adaptive Server のすべてのユーザが使用できるロックの総数を設定します。

Adaptive Server が必要とするロックの総数は、同時プロセスと並列プロセスの数、およびトランザクションが実行するアクションのタイプによって異なります。ある時点でどれくらいのロックが使用されているかを調べるには、sp_lock を使用します。

逐次処理の場合は、初めはアクティブな同時接続 1 つ当たり 20 のロック数を割り当てることをおすすめします。

並列処理に必要なロックの数は逐次処理よりも多くなります。たとえば、クエリが平均で 5 つのワーカー・プロセスを使用することがわかった場合は、逐次処理に対して設定した number of locks の値を 1/3 ずつ増やしてみてください。

使用できるロックが足りなくなると、Adaptive Server は、サーバ・レベルのエラー・メッセージを表示します。ユーザがロック・エラーを報告する場合は、number of locks の値を大きくする必要があるかもしれません。ただし、ロックはメモリを消費することを念頭においてください。『システム管理ガイド 第 2 巻』の「第 3 章 メモリの設定」を参照してください。

注意 データロー・ロックを使用する場合は、number of locks の値を変更する必要があります。詳細については、『パフォーマンス＆チューニング・シリーズ：ロックと同時実行制御』を参照してください。

number of mailboxes

要約	
デフォルト値	30
値の範囲	30 ～ 2147483647
ステータス	動的
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	メモリ使用、SQL Server 管理

number of mailboxes パラメータは、Adaptive Server が割り付けるメールボックス構造体の数を指定します。メールボックスはメッセージとともに使用され、カーネル・サービス・プロセス間の通信と同期用に内部的に使用されます。ユーザ・プロセスはメールボックスを使用しません。Sybase 製品の保守契約を結んでいるサポート・センタから指示がないかぎり、このパラメータは変更しないでください。

number of messages

要約	
デフォルト値	64
値の範囲	0 ～ 2147483647
ステータス	動的
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	メモリ使用、SQL Server 管理

number of messages パラメータは、Adaptive Server が割り付けるメッセージ構造体の数を指定します。メッセージはメールボックスとともに使用され、カーネル・サービス・プロセス間の通信と同期用に内部的に使用されます。また、メッセージは、並列処理におけるプロセスのファミリ間の調整にも使用されます。Sybase 製品の保守契約を結んでいるサポート・センタから指示がないかぎり、このパラメータは変更しないでください。

number of oam trips

要約	
デフォルト値	0
値の範囲	0 ～ 65535
ステータス	動的
表示レベル	包括
必要な役割	システム管理者

number of oam trips は、古くなった **object allocation map** (OAM) ページが、何回 MRU/LRU チェーンを周回してから追い出されるかを指定します。number of oam trips の値を増やすに従って、古くなった OAM ページがキャッシュ内にとどまる時間が長くなります。

テーブルやインデックスごとに OAM ページがあり、それぞれの OAM ページにはテーブルやインデックスに割り当てられたページに関する情報が格納され、インデックスやテーブルに新しいページが必要になった場合にチェックされます (「page utilization percent」(205 ページ) を参照)。1 つの OAM ページは、2,000 ～ 63,750 のデータまたはインデックス・ページ用のアロケーション・マップを保持できます。

OAM ページは、オブジェクトが領域を使用するそれぞれのアロケーション・ユニットのアロケーション・ページを指します。アロケーション・ページには、アロケーション・ユニット内のエクステントとページの使用状況についての情報が記録されています。

大量のバルク・コピー操作のように非常に大きな領域の割り付けを必要とする環境やベンチマークでは、OAM ページがキャッシュ内に存在する時間が長いほどパフォーマンスが向上します。number of oam trips を大きな値に設定するほど、OAM ページをキャッシュ内に長くとどめておくことができます。

注意 キャッシュが比較的小さく、数多くのオブジェクトによって使用される場合は、number of oam trips をあまり大きな値に設定しないでください。設定値が大きすぎると、追い出されない OAM ページでキャッシュがいっぱいになり、ユーザ・スレッドのタイムアウトが発生する可能性があります。

number of oam trips の値を 0 以外に変更する前に、すべてのインデックス、OAM、データ・ページを保管できるだけの十分なキャッシュがアプリケーションにあることを確認してください。number of oam trips の値を変更する前に、Sybase 製品の保守契約を結んでいるサポート・センタまで連絡してください。

number of open databases

要約	
デフォルト値	12
値の範囲	6 ～ 2147483647
ステータス	動的
表示レベル	基本
必要な役割	システム管理者
設定グループ	メモリ使用、メタデータ・キャッシュ、SQL Server 管理

number of open databases は、Adaptive Server で同時にオープンできるデータベースの最大数を指定します。

値を計算するときは、システム・データベース master、model、sybsystemprocs、tempdb も加えてください。監査機能をインストールした場合は、sybsecurity データベースも加えます。また、サンプル・データベース pubs2 と pubs3、構文データベース sybsyntax、dbcc データベース dbccdb をインストールした場合は、これらも加えてください。

別のサーバから大きなデータベースをロードするなどの大幅な変更を行おうとしている場合は、sp_helpconfig を使用してメタデータ・キャッシュ・サイズの概算値を計算します。sp_helpconfig は、特定の数のメタデータ記述子に必要なメモリ量と、特定のメモリ量で対応可能な記述子の数を表示します。データベース・メタデータ記述子は、データベースの使用中の状態、またはキャッシュされている状態を表します。

❖ ***number of open databases* の最適化**

Adaptive Server でオープン可能なデータベースの数を超えたことを知らせるメッセージが表示された場合は、この値を調整してください。

1　sp_countmetadata を使用して、データベース・メタデータ記述子の総数を調べます。

```
sp_countmetadata "open databases"
```

sp_countmetadata の実行に最適なのは、サーバ上にアクティビティがほとんどないときです。ピーク時に sp_countmetadata を実行すると、他のプロセスとの競合が発生します。

Adaptive Server から次の情報がレポートされたとします。

```
There are 50 databases, requiring 1719 Kbytes of memory.
The 'open databases' configuration parameter is currently
set to 500.
```

2　number of open databases の値を 50 に設定します。

```
sp_configure "number of open databases", 50
```

この新しい設定値は、最終的な値ではありません。データベースの総数ではなく、アクティブなデータベースのメタデータ・キャッシュ記述子の数を基準に設定してください。

3　ピーク時にアクティブなメタデータ記述子の数を調べます。

```
sp_monitorconfig "open databases"

Usage Information at date and time: Apr 22 2002  2:49PM.
Name              num_free    num_active   pct_act     Max_Used   Reused
--------------    --------    ---------   --------    --------   ------
number of open    50          20           40.00       26         No
```

この例では、20 のメタデータ・データベース記述子がアクティブになっています。サーバの最後の起動後にアクティブになった記述子の最大数は 26 です。

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_monitorconfig」を参照してください。

4　number of open databases を、26 に 10% (およそ 3) の追加領域を加えた値、つまり 29 に設定します。

```
sp_configure "number of open databases", 29
```

たとえば、データベースの追加や削除の操作を行っているときなどに、サーバ上に多くのアクティビティがある場合は、sp_monitorconfig を定期的に実行します。アクティブな記述子の数の変化に応じて、キャッシュ・サイズを再設定してください。

number of open indexes

要約	
デフォルト値	500
値の範囲	100 ～ 2147483647
ステータス	動的
表示レベル	基本
必要な役割	システム管理者
設定グループ	メモリ使用、メタデータ・キャッシュ

number of open indexes パラメータは、Adaptive Server で同時に使用できるインデックスの最大数を設定します。

大量のインデックスを持つデータベースを別のサーバからロードするなどの大幅な変更を行おうとしている場合に、sp_helpconfig を使用すると、メタデータ・キャッシュ・サイズの概算値を計算できます。sp_helpconfig は、特定の数のメタデータ記述子に必要なメモリ量と、特定のメモリ量で対応可能な記述子の数を表示します。インデックス・メタデータ記述子は、インデックスの使用中の状態、またはキャッシュされている状態を表します。

❖ ***number of open indexes* の最適化**

number of open indexes のデフォルト値では不十分な場合、Adaptive Server はアクティブなインデックス記述子の再使用を試みた後でメッセージを表示します。このメッセージが表示された場合は、値を調整する必要があります。

1 sp_countmetadata を使用して、インデックス・メタデータ記述子の総数を調べます。

```
sp_countmetadata "open indexes"
```

sp_countmetadata の実行に最適なのは、サーバ上のアクティビティがほとんどないときです。ピーク時に sp_countmetadata を実行すると、他のプロセスとの競合が発生します。

Adaptive Server から次の情報がレポートされたとします。

```
There are 698 user indexes in all database(s),
requiring 286.289 Kbytes of memory. The 'open
indexes' configuration parameter is currently set to
500.
```

2 number of open indexes パラメータを 698 に設定します。

```
sp_configure "number of open indexes", 698
```

この新しい設定は最終的な値ではありません。インデックスの総数ではなく、アクティブなインデックスのメタデータ・キャッシュ記述子の数を基準に設定してください。

3　ピーク時に、アクティブなインデックス・メタデータ記述子の数を調べます。

```
sp_monitorconfig "open indexes"

Usage Information at date and time: Apr 22 2002  2:49PM.
Name              num_free    num_active   pct_act     Max_Used   Reused
--------------    --------    ---------   --------    --------   ------
number of open     182         516         73.92       590        No
```

この例では、サーバの最後の起動後に使用されたインデックス記述子の最大数は 590 です。

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

4　number of open indexes 設定パラメータを、590 に 10% の追加領域 (59) を加えた値、つまり 649 に設定します。

```
sp_configure "number of open indexes", 649
```

たとえば、テーブルの追加や削除などの操作を行っているときなどに、サーバ上に多くのアクティビティがある場合は、sp_monitorconfig を定期的に実行します。アクティブな記述子の数の変化に応じて、キャッシュ・サイズを再設定してください。

number of open objects

要約	
デフォルト値	500
値の範囲	100 ～ 2147483647
ステータス	動的
表示レベル	基本
必要な役割	システム管理者
設定グループ	メモリ使用、メタデータ・キャッシュ、SQL Server 管理

number of open objects は、Adaptive Server で同時にオープンできるオブジェクトの最大数を指定します。

大量のオブジェクトが含まれるデータベースを別のサーバからロードするなどの大幅な変更を行おうとしている場合に、sp_helpconfig を使用するとメタデータ・キャッシュ・サイズの概算値を計算できます。sp_helpconfig は、特定の数のメタデータ記述子に必要なメモリ量と、特定のメモリ量で対応可能な記述子の数を表示します。オブジェクト・メタデータ記述子は、インデックスの使用中の状態、またはキャッシュされている状態を表します。

❖ ***number of open objects* の最適化**

number of open objects のデフォルト値では不十分な場合、Adaptive Server はアクティブなオブジェクト記述子の再使用を試みた後でメッセージを表示します。

1 システム・プロシージャ sp_countmetadata を使用して、オブジェクト・メタデータ記述子の総数を調べます。

```
sp_countmetadata "open objects"
```

sp_countmetadata の実行に最適なのは、サーバ上のアクティビティがほとんどないときです。ピーク時に sp_countmetadata を実行すると、他のプロセスとの競合が発生します。

Adaptive Server から次の情報がレポートされたとします。

```
There are 1340 user objects in all database(s), requiring
1443 Kbytes of memory. The 'open objects' configuration
parameter is currently set to 500.
```

2 オープンしているオブジェクトの数を占めるように number of open objects を設定します。

```
sp_configure "number of open objects", 1407
```

1407 は、ユーザ・オブジェクト数 1340 に、テンポラリ・テーブル用の 5% を加えた値です。

この新しい設定は最終的な値ではありません。理想的な数は、オブジェクトの総数ではなく、アクティブなオブジェクトのメタデータ・キャッシュ記述子の数を基準に設定してください。

3 ピーク時に、アクティブなメタデータ・キャッシュ記述子の数を調べます。

```
sp_monitorconfig "open objects"

Usage Information at date and time: Aug 20 2007  1:32PM..
Name                     Num_free   Num_active    Pct_act     Max_Used
Num_reuse
--------------           --------    ---------    --------    --------
------
number of open objects       560          847       71.40        1397
0
```

この例では、サーバの最後の起動後に使用されたオブジェクト記述子の最大数は 1497 です。

4 number of open objects を、1397 に 10% の追加領域 (140) を加えた値、つまり 1537 に設定します。

```
sp_configure "number of open objects", 1537
```

たとえば、テーブルの追加や削除などの操作を行っているときなどに、サーバ上に多くのアクティビティがある場合は、sp_monitorconfig を定期的に実行します。アクティブな記述子の数の変化に応じて、キャッシュ・サイズを再設定してください。『リファレンス・マニュアル：プロシージャ』の「sp_monitorconfig」を参照してください。

number of open partitions

要約	
デフォルト値	500
値の範囲	100 ～ 2147483647
ステータス	動的
表示レベル	基本
必要な役割	システム管理者
設定グループ	メモリ使用、メタデータ・キャッシュ

Adaptive Server が一度にアクセスできるパーティションの数を指定します。

使用システムに合わせた *number of open partitions* パラメータの最適化

number of open partitions のデフォルト値では不十分な場合、Adaptive Server はアクティブなパーティション記述子の再使用を試みた後でメッセージを表示します。このメッセージが表示された場合は、値を調整する必要があります。

1　システム・プロシージャ sp_countmetadata を使用して、開いているパーティションの総数を調べます。次に例を示します。

```
sp_countmetadata "open partitions"
```

sp_countmetadata の実行に最適なのは、サーバ上のアクティビティがほとんどないときです。ピーク時に sp_countmetadata を実行すると、他のプロセスとの競合が発生します。

Adaptive Server から次の情報がレポートされたとします。

```
There are 42 user partitions in all database(s),
requiring 109 Kbytes of memory. The 'open
partitions' configuration parameter is currently set
to 110.
```

2　sp_countmetadata の報告に従って、number of open partitions を 110 に設定します。

```
sp_configure "number of open partitions", 110
```

3 ピーク時に、アクティブなメタデータ・キャッシュ記述子の数を調べます。次に例を示します。

```
sp_monitorconfig "open partitions"
Usage Information at date and time: Jun 30 2008  3:15PM.

Name                          Num_free         Num_active      Pct_act
Max_Used        Reuse_cnt
----------------------------  --------------   ---------       --------
-----------     ---------
number of open partitions     27               57              51.8
83              0
```

この例では、サーバの最後の起動後に使用されたパーティションの最大数は 83 です。

4 number of open partitions パラメータを、83 に 10% の追加領域 (8) を加えた値、つまり 91 に設定します。

```
sp_configure "number of open partitions", 91
```

たとえば、テーブルの追加や削除などの操作を行っているときなどに、サーバ上に多くのアクティビティがある場合は、sp_monitorconfig を定期的に実行します。アクティブな記述子の数の変化に応じて、キャッシュ・サイズを再設定してください。『リファレンス・マニュアル：プロシージャ』の「sp_monitorconfig」を参照してください。

number of pre-allocated extents

要約	
デフォルト値	2
値の範囲	1 ～ 32
ステータス	動的
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	SQL Server 管理

number of pre-allocated extents は、ページ・マネージャへの 1 回の周回で割り付けられるエクステント (8 ページ) の数を指定します。現在、このパラメータを使用するのは bcp だけで、その目的は大量のデータをコピーするときのパフォーマンスの改善です。デフォルトでは、bcp は一度に 2 つのエクステントを割り付け、そのたびに割り付けレコードをログに書き込みます。

number of pre-allocated extents を設定することは、bcp がより多くの領域を必要とするたびに指定された数のエクステントを割り付け、そのイベントについて 1 つのログ・レコードを書き込むことを意味します。

実際に必要とするよりも多くのページがオブジェクトに割り付けられることがあるため、bcp で処理するバッチが小さい場合は、number of pre-allocated extents の値を小さくしてください。bcp で処理するバッチが大きい場合は、number of pre-allocated extents の値を大きくして、ページの割り付けに必要なオーバヘッドの量とログ・レコード数を減らしてください。

number of pre-allocated extents への値 32 の使用

number of pre-allocated extents に値 32 を使用することは、設定にとって特別な意味があり、Adaptive Server が内部的に実行する領域の割り付けに影響を及ぼします。number of pre-allocated extents が 32 に設定されると、Adaptive Server はエクステント数分の割り付け単位全体を bcp-in や select into などのユーティリティ操作のために予約します。これらの操作では、領域予約の大規模な割り付けスキームが使用されるからです。これにより、これらのユーティリティのパフォーマンスが大幅に向上します。特に、複数のノード上で同時に実行する場合です。したがって、値 32 を使用すると、クラスタの各ノードが独自の割り付け単位上で他のノードから干渉されることなく独立して作業できます。

以前のバージョンの Adaptive Server では、number of pre-allocated extents パラメータは、あらゆるサイズのテーブル向けに単一の割り付け呼び出しで予約されるエクステント数を指定していました。

このバージョンの Adaptive Server では、次のコマンドの場合にかぎり、ページが 240 行以上ある大規模なテーブルでは number of pre-allocated extents の値が無視されます。

- alter table *table_name* add *column_name* . . .
- alter table *table_name* modify *column_name* . . .
- alter table *table_name* drop *column_name* . . .
- alter table lock . . .
- reorg rebuild

これらのコマンドを 240 ページを超えるテーブルで実行するときには、Adaptive Server は割り付け単位全体 (エクステント数 32) を予約します。これにより、パフォーマンスが大幅に向上します。特に、複数のノードで同時に実行するときです。

number of pre-allocated extents の値は、240 ページ未満のテーブルに対する上記のコマンドだけでなく、あらゆるサイズのテーブルに対するすべてのコマンド (select into、bcp、alter table partition など) でも引き続き守られます。

number of Q engines at startup

要約	
デフォルト値	0
値の範囲	0 ～ 127
ステータス	静的
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	プロセッサ

number of Q engines at startu は、サーバ起動時にオンラインになる Q エンジンの数を指定し、MQ に必要です。max online Q engines の数値に対応するには、max online engines を増やす必要がある場合があります。

number of remote connections

要約	
デフォルト値	20
値の範囲	5 ～ 32767
ステータス	静的
表示レベル	中間
必要な役割	システム管理者
設定グループ	メモリ使用、ネットワーク通信

number of remote connections は、1 つの Adaptive Server との間で同時にオープンできる論理接続数を指定します。ESP 実行のための XP Server への同時接続には、それぞれ 1 つのリモート接続を使用します。「第 15 章 リモート・サーバの管理」を参照してください。

number of remote logins

要約	
デフォルト値	20
値の範囲	0 ～ 32767
ステータス	静的
表示レベル	中間
必要な役割	システム管理者
設定グループ	メモリ使用、ネットワーク通信

number of remote logins パラメータは、Adaptive Server からリモート・サーバへのアクティブなユーザ接続数を制御します。ESP 実行のための XP Server への同時接続には、それぞれ 1 つのリモート接続を使用します。このパラメータの値は、number of remote connections の値以下となるように設定してください。「第 15 章 リモート・サーバの管理」を参照してください。

number of remote sites

要約	
デフォルト値	10
値の範囲	0 ～ 32767
ステータス	静的
表示レベル	中間
必要な役割	システム管理者
設定グループ	メモリ使用、ネットワーク通信

number of remote sites は、Adaptive Server に同時にアクセスできるリモート・サイトの最大数を決定します。Adaptive Server と XP Server 間の接続にはそれぞれ 1 つのリモート・サイト接続を使用します。

内部的には、number of remote sites によって、一度にアクティブにできるサイト・ハンドラの数が決定します。同じサイトからのサーバ・アクセスはすべて 1 つのサイト・ハンドラによって管理されます。たとえば、number of remote sites が 5 に設定されているときに、各サイトが 3 つずつリモート・プロシージャ・コールを実行すると、sp_who の出力には、15 個のプロセスに対する 5 つのサイト・ハンドラ・プロセスがあることが表示されます。「第 15 章 リモート・サーバの管理」を参照してください。

number of sort buffers

要約	
デフォルト値	500
値の範囲	0 ～ 32767
ステータス	動的
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	SQL Server 管理

number of sort buffers は、ソート処理時に入力テーブルから読み込んだページを保持し、インデックスのマージを実行するために使用するバッファに割り付けられるメモリの量を指定します。並列ソートは次の場合に使用されます。

- updates statistics の実行
- インデックスの作成

『パフォーマンス&チューニング・シリーズ：クエリ処理と抽象プラン』の「第 10 章 パフォーマンス改善のための統計値の使用」を参照してください。

number of sort buffers の値は、サーバのページ・サイズによって異なります。

このパラメータの値は、インデックスを並列に作成する場合を除いて、デフォルトの設定のままにしておくことをおすすめします。

設定値が大きすぎると、ソートを実行するために使用されるキャッシュ内のバッファ・プールに、ソート以外のプロセスがアクセスできなくなります。

ソート・バッファに大きな値を設定した場合、大きなテーブルのソートではプロシージャ・キャッシュがさらに必要になる可能性があります。この影響は、ローのサイズが小さいテーブルのほうが顕著になります。これは 1 ページあたりのローの数が多くなるためです。

次の式で、必要なプロシージャ・キャッシュの量 (バイト単位) を概算できます。

(ソート・バッファ数) x (*1* ページ当たりのロー数) x 100

ソート・バッファの数に対してプロシージャ・キャッシュが不足している場合、ソートは失敗し、エラー・メッセージ 701 が表示されることがあります。このメッセージが表示された場合は、ソート・バッファの設定数を小さくしてからソートを再実行してください。

1 つの演算子が使用するバッファ数の上限値を設定する方法については、「max buffers per lava operator」(165 ページ) を参照してください。

number of user connections

要約	
デフォルト値	25
値の範囲	5 ～ 2147483647
ステータス	動的
表示レベル	基本
必要な役割	システム管理者
設定グループ	メモリ使用、ユーザ環境

number of user connections は、同時に Adaptive Server へ接続できるユーザ接続の最大数を設定します。これはプロセスの最大数を指すものではありません。プロセスの最大数はこのパラメータの値だけでなく、他のシステム・アクティビティにも依存します。

最大ユーザ接続数の上限値

プロセス当たりのファイル記述子の最大許容数は、オペレーティング・システムによって異なります。プラットフォームの『Adaptive Server Enterprise 設定ガイド』を参照してください。

Adaptive Server 接続で使用可能なファイル記述子の数は、グローバル変数 *@@max_connections* に格納されています。使用しているシステムで使用可能なファイル記述子の最大数をレポートするには、次のようにします。

```
select @@max_connections
```

戻り値は、システムによってプロセスに許されるファイル記述子の最大数から、オーバヘッドを差し引いた数を表します。オーバヘッドは、エンジン数とともに増加します。Adaptive Server 接続で使用可能なファイル記述子の数に対するマルチプロセッシングの影響については、『システム管理ガイド 第 2 巻』の「第 5 章 マルチプロセッサ・サーバの管理」を参照してください。

さらに、次の項目の分の接続を確保する必要があります。これらの項目も、設定パラメータを使用して設定されます。

- データベース・デバイス (ミラー・デバイスを含む)
- サイト・ハンドラ
- ネットワーク・リスナ

number of user connections + (number of devices * max online engines * 2) + number of remote sites + max number network listeners が *@@max_connections* の値を超えないようにします。

予約済みの接続

設定された接続数のうち 1 つは、一時的な管理タスク用に予約されます。これは、データベース管理者が必ず Adaptive Server に接続できるようにするためです。予約済みの接続では、ログインできる時間は合計で 15 分であり、sa_role を持つユーザにのみ割り付けられます。15 分が経過すると接続は終了します。これは、複数のデータベース管理者がいるインストール環境で他の管理者が予約済みの接続を使用できるようにするためです。

Adaptive Server に接続するための最後のリソースがクライアントによって使用されているときも、この予約済みの接続が自動的に使用されます。

Adaptive Server が予約済みの接続を使用しているときにユーザが Adaptive Server にログインすると、次のメッセージが表示されます。

```
There are not enough user connections available; you are being connected
using a temporary administrative connection which will time out after '15'
minutes. Increase the value of th 'number of user connections' parameter
```

また、Adaptive Server への最後の接続がタイムアウトで切断されたときは、次のようなメッセージがエラー・ログに出力されます。

```
00:00000:00008:2003/03/14 11:25:31.36 server  Process '16' has been
terminated as it exceeded the maximum login time allowed for such processes.
This process used a connection reserved for system administrators and has a
maximum login period of '15' minutes
```

***max number of user connections* の最適化**

それぞれのユーザに許可する接続数を決定するための公式はありません。システムとユーザの要件に基づいてこの値を見積もってください。ユーザ数の多いシステムでは、時々または一時的にしか必要とされない接続は、一般に複数のユーザ間で共有できることも考慮に入れてください。ユーザ接続を必要とするプロセスは次のとおりです。

- isql を実行するユーザごとに 1 つの接続が必要となる。
- アプリケーション開発者は、編集セッションごとに 1 つの接続を使用する。

- アプリケーションを実行するユーザが必要とする接続数は、そのアプリケーションがどのようにプログラムされているかに依存する。Open Client プログラムを実行するユーザは、オープン DB-Library dbprocess または Client-Library™ cs_connection ごとに 1 つの接続を必要とする。

注意 Adaptive Server によって使用される接続の最大数を見積もり、システムに物理デバイスまたはユーザを追加するときに number of user connections を更新することをおすすめします。sp_who を使用して、Adaptive Server 上のアクティブなユーザ接続数を定期的に調べてください。

この他に、stack size や default network packet size などの設定パラメータもユーザ接続ごとのメモリ量に影響を与えます。

共有メモリへのユーザ接続 − EJB サーバ

Adaptive Server は、number of user connections の値を使用して、EJB サーバの共有メモリ接続の数を決定します。したがって、number of user connections が 30 の場合、Adaptive Server は、EJB サーバ用に 10 個の共有メモリ接続を確立します。共有メモリ接続はユーザ接続のサブセットではないので、ユーザ接続数から減算されません。

共有メモリのユーザ接続数を増やすには、次の手順に従います。

1 number of user connections を、必要な共有メモリ接続数の 3 倍に増やします。

2 Adaptive Server を再起動します。

number of user connections は動的な設定パラメータですが、共有メモリ用のユーザ接続数を変更するにはサーバを再起動する必要があります。『EJB Server ユーザーズ・ガイド』を参照してください。

Adaptive Server バージョン 12.5.3 ESD #2 では、EJB のためにソケットが自動的に予約されることはありません。ただし、トレース・フラグ 1642 を有効にすることで、以前のバージョンの機能に戻してソケットの 3 分の 1 を EJB 用に予約できます。EJB サーバをセットアップするには、トレース・フラグ 1642 を有効にします。Adaptive Server のこのバージョンでは、EJB サーバが設定されていなければ、エラー・ログに記録された "`hbc_ninit: No sockets available for HBC`" というメッセージを無視できます。

Adaptive Server バージョン 12.5.3 とそれ以降では、EJB サーバは有効になっているにもかかわらず、HBC ソケットが使用できない場合は "`hbc_ninit: No sockets available for HBC`" というメッセージが報告されます。トレース・フラグ 1642 が有効ではない場合は、フラグを設定し、Adaptive Server を再起動します。EJB サーバが有効でない場合はメッセージは報告されず、EJB サーバ用に予約されたソケットは自動的に無効になります。

number of worker processes

要約	
デフォルト値	0
値の範囲	0 ～ 2147483647
ステータス	動的
表示レベル	基本
必要な役割	システム管理者
設定グループ	メモリ使用、クエリ・チューニング

number of worker processes は、同時に実行されるすべての並列クエリ全体に対して Adaptive Server が使用できるワーカー・プロセスの最大数を指定します。

メモリ不足のため、指定された数のワーカー・プロセスを作成できない場合は、Adaptive Server の起動時に警告メッセージが表示されます。memory per worker process は、ワーカー・プロセスごとに割り付けられるメモリを制御します。

number of worker processes をワーカー・スレッド・プールの十分な数のスレッド数に設定していない場合、Adaptive Server はより少ない数のワーカー・スレッドを使用するように実行時にクエリ・プランを調整します。Adaptive Server が実行時にクエリを調整できない場合、クエリは逐次モードで再コンパイルされます。ただし、十分な数のワーカー・スレッドがない場合、alter table および execute immediate コマンドは中止されます。

o/s file descriptors

要約	
デフォルト値	0
値の範囲	サイト固有
ステータス	読み込み専用
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	O/S リソース

o/s file descriptors は、オペレーティング・システムに設定されている、プロセス当たりのファイル記述子の最大数を表します。このパラメータは読み込み専用であり、Adaptive Server からの設定はできません。

多くのオペレーティング・システムでは、プロセスごとに使用できるファイル記述子の数を設定できます。使用するオペレーティング・システム用のマニュアルを参照してください。

Adaptive Server 接続で使用できるファイル記述子の数は、変数 *@@max_connections* に格納されています。この数は o/s file descriptors の値よりも小さくなります。「最大ユーザ接続数の上限値」(196 ページ) を参照してください。

object lockwait timing

要約	
デフォルト値	0 (オフ)
値の範囲	0 (オフ)、1 (オン)
ステータス	動的
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	モニタリング

object lockwait timing は、Adaptive Server でオブジェクトに対するロック要求のタイミング統計を収集するかどうかを制御します。

open index hash spinlock ratio

要約	
デフォルト値	100
値の範囲	1 ～ 2147483647
ステータス	動的
表示レベル	基本
必要な役割	システム管理者
設定グループ	メモリ使用、メタデータ・キャッシュ

open index hash spinlock ratio は、1 つの **spinlock** によって保護されるインデックス・メタデータ記述子のハッシュ・テーブル数を設定します。このパラメータは、マルチプロセッシング・システムでのみ使用します。

テーブルに属しているすべてのインデックス記述子は、ハッシュ・テーブルを通じてアクセスできます。テーブルに対するクエリが実行されると、Adaptive Server はハッシュ・テーブルを使用し、必要なインデックス情報をその sysindexes ロー内で検索します。ハッシュ・テーブルは、情報をすばやく取り出すために Adaptive Server が使用する内部メカニズムです。

通常、このパラメータを変更する必要はありません。ただし、ごくまれに、ハッシュ・スピンロックからの競合が発生している場合に、このパラメータを再設定する必要があります。『パフォーマンス＆チューニング・シリーズ：sp_sysmon による Adaptive Server の監視』を参照してください。

スピンロック率の設定の詳細については、『システム管理ガイド 第 2 巻』の「第 5 章 マルチプロセッサ・サーバの管理」を参照してください。

open index spinlock ratio

要約	
デフォルト値	100
値の範囲	1 ～ 214748364
ステータス	動的
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	メモリ使用、メタデータ・キャッシュ

open index spinlock ratio は、1 つの **spinlock** によって保護されるインデックス・メタデータ記述子の数を設定します。

インデックス記述子の内容に複数のプロセスがアクセスすることがあるので、Adaptive Server はスピンロックを使用してインデックス記述子を保護します。open index spinlock ratio は、マルチプロセッシング・システムでのみ使用します。

このパラメータに指定する値は、スピンロック当たりのインデックス記述子の比率を定義します。

1 つのスピンロックを共有するインデックス記述子が多すぎると、スピンロック競合が発生する場合があります。スピンロック競合のレポートを取得するには、sp_sysmon を使用します。『パフォーマンス＆チューニング・シリーズ：sp_sysmon による Adaptive Server の監視』を参照してください。

sp_sysmon で出力されるインデックス記述子のスピンロック競合が 3% を超えている場合は、open index spinlock ratio の値を小さくしてみてください。

『システム管理ガイド 第 2 巻』の「第 5 章 マルチプロセッサ・サーバの管理」を参照してください。

open object spinlock ratio

要約	
デフォルト値	100
値の範囲	1 ～ 2147483647
ステータス	動的
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	メタデータ・キャッシュ

open object spinlock ratio は、1 つの **spinlock** によって保護されるオブジェクト記述子の数を設定します。オブジェクト記述子の内容に複数のプロセスがアクセスすることがあるので、Adaptive Server はスピンロックを使用してオブジェクト記述子を保護します。open object spinlock ratio は、マルチプロセッシング・システムでのみ使用します。

このパラメータのデフォルト値は 100 です。つまり、サーバに設定されているオブジェクト記述子 100 個につき 1 個のスピンロックがあることになります。サーバに設定されているエンジンが 1 つだけの場合は、オブジェクト記述子の数に関係なく、設定されるオブジェクト記述子のスピンロックは 1 つだけです。

1 つのスピンロックを共有するオブジェクト記述子が多すぎる場合は、スピンロック競合が発生します。スピンロック競合のレポートを取得するには、sp_sysmon を使用します。『パフォーマンス＆チューニング・シリーズ：sp_sysmon による Adaptive Server の監視』を参照してください。

sp_sysmon で出力されるオブジェクト記述子のスピンロック競合が 3% を超えている場合は、open object spinlock ratio パラメータの値を小さくしてみてください。

『システム管理ガイド 第 2 巻』の「第 5 章 マルチプロセッサ・サーバの管理」を参照してください。

optimization goal

要約	
デフォルト値	allrows_mix
値の範囲	allrows_oltp、allrows_dss
ステータス	動的
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	クエリ・チューニング

最適化目標は、最も優れた最適化テクニックを使用してクエリ要求を満たす便利な方法であり、オプティマイザの時間とリソースの最適利用を保証します。Adaptive Server では、2 つの最適化目標を、サーバ・レベル、セッション・レベル、クエリ・レベルという 3 つの層で指定できます。

サーバ・レベルの最適化目標はセッション・レベルの最適化目標によって上書きされ、セッション・レベルの最適化目標はクエリ・レベルの最適化目標によって上書きされます。

次の最適化目標を設定することで、各自のクエリ環境に最も適した最適化方式を選択できます。

- allrows_oltp － OLTP クエリにとって最も有用な目標です。
- allrows_dss － 中程度から高度に複雑な業務的 DSS クエリにとって最も有用な目標です。

optimization timeout limit

要約	
デフォルト値	10
値の範囲	0 ～ 1000
ステータス	動的
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	クエリ・チューニング

optimization timeout limit は、Adaptive Server がクエリを最適化するために費やすことができる時間を、クエリの推定実行時間に対する割合として指定します。

値 0 は、最適化のタイムアウトがないことを示します。

page lock promotion HWM

要約	
デフォルト値	200
値の範囲	2 ～ 2147483647
ステータス	動的
表示レベル	中間
必要な役割	システム管理者
設定グループ	ロック・マネージャ、SQL Server 管理

page lock promotion HWM (high-water mark) パラメータは、page lock promotion LWM (low-water mark) と page lock promotion PCT (percentage) とともに使用します。このパラメータは、ページがロックされるテーブルまたはインデックスの 1 回のスキャン・セッション中に許容されるページ・ロック数を指定します。この数に達すると、Adaptive Server はページ・ロックからテーブル・ロックへの拡大を試みます。

1 回のスキャン・セッション中に取得したページ・ロックの数が page lock promotion HWM を超えると、Adaptive Server はテーブル・ロックを取得しようとします。page lock promotion HWM の値は、number of locks の値より大きくすることはできません。

スキャン・セッションの説明と、ページ・ロック・プロモーションの制限値の設定方法の詳細については、『パフォーマンス&チューニング・シリーズ：ロックと同時実行制御』の「第 2 章 ロックの設定とチューニング」を参照してください。

page lock promotion HWM のデフォルト値は、ほとんどのアプリケーションにとって適切な値です。テーブルのロックを回避するには、値を大きくします。たとえば、数千ものページがある全ページロック・テーブルまたはデータページロック・テーブルのうち 500 ページに対して定期的な更新が行われることがわかっている場合は、page lock promotion HWM を 500 に設定して、これらのテーブルの同時実行性を高めます。

また、ページロック・テーブルとビューのロック・プロモーションを、オブジェクト単位のレベルで設定することもできます。『リファレンス・マニュアル：プロシージャ』の「sp_setrowlockpromote」を参照してください。

page lock promotion HWM パラメータの変更がロック拡大の数にどのように影響するかを確認するには、sp_sysmon を使用してください。sp_sysmon は、排他ページ・ロックから排他テーブル・ロックへのプロモーションの比率と、共有ページ・ロックから共有テーブル・ロックへのプロモーションの比率をレポートします。『パフォーマンス&チューニング・シリーズ：sp_sysmon による Adaptive Server の監視』を参照してください。

page lock promotion LWM

要約	
デフォルト値	200
値の範囲	2 ～ page lock promotion HWM の値
ステータス	動的
表示レベル	中間
必要な役割	システム管理者
設定グループ	ロック・マネージャ、SQL Server 管理

page lock promotion LWM (low-water mark) は、page lock promotion HWM (high-water mark) と page lock promotion PCT (percentage) とともに使用します。このパラメータは、ページがロックされるテーブルやインデックスの 1 回のスキャン・セッション中に許容されるページ・ロック数を指定します。この数に達すると、Adaptive Server はページ・ロックからテーブル・ロックへの拡大を試みます。

page lock promotion LWM で設定されたページ・ロック数に達するまでは、Adaptive Server はオブジェクトに対するテーブル・ロックを発行しません。page lock promotion LWM には、page lock promotion HWM 以下の値を設定してください。

page lock promotion LWM のデフォルト値は、ほとんどのアプリケーションにとって適切な値です。Adaptive Server がロックを使い果たす状況が繰り返し発生する場合は、number of locks を増やしてください。

詳細については、『パフォーマンス&チューニング・シリーズ：ロックと同時実行制御』を参照してください。

ページ・ロック・プロモーションは、オブジェクト単位のレベルで設定することもできます。『リファレンス・マニュアル：プロシージャ』の「sp_setpglockpromote」を参照してください。

page lock promotion PCT

要約	
デフォルト値	100
値の範囲	1 ～ 100
ステータス	動的
表示レベル	中間
必要な役割	システム管理者
設定グループ	ロック・マネージャ、SQL Server 管理

オブジェクトに対して保持されているロック数が page lock promotion LWM (low-water mark) と page lock promotion HWM (high-water mark) の間の値である場合に、page lock promotion PCT で設定されたページ・ロックのパーセンテージ (テーブル・サイズに基づく) を超えると、Adaptive Server はテーブル・ロックの取得を試みます。

『パフォーマンス＆チューニング・シリーズ：ロックと同時実行制御』の「第 2 章 ロックの設定とチューニング」を参照してください。

page lock promotion PCT のデフォルト値は、ほとんどのアプリケーションに適切な値です。

ページがロックされるオブジェクトに対するロック・プロモーションは、オブジェクト単位のレベルで設定することもできます。『リファレンス・マニュアル：プロシージャ』の「sp_setpglockpromote」を参照してください。

page utilization percent

要約	
デフォルト値	95
値の範囲	1 ～ 100
ステータス	動的
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	ディスク I/O

page utilization percent はページを割り付けるときに使用され、Adaptive Server がテーブルのオブジェクト・アロケーション・マップ (OAM: Object Allocation Map) をスキャンして未使用のページを検索するか、または単純に新しいエクステントをテーブルに割り付けるかを制御します。OAM の詳細については、“number of oam trips” (185 ページ) を参照してください。page utilization percent パラメータは、大規模なテーブルを持つサーバのパフォーマンスを最適化し、新しいスペースを追加する時間を短縮します。

page utilization percent が 100 に設定されているときは、Adaptive Server は新しいエクステントを割り付ける前に、オブジェクトに割り付けられた未使用のページを見つけるためにすべての OAM ページをスキャンします。このパラメータの設定値が 100 より小さい場合は、page utilization percent の設定と、次の計算式で求めたテーブルの使用済みページ数 (used pages) と未使用ページ数 (unused pages) の比とを比較します。

```
100 * used pages/(used pages + unused pages)
```

page utilization percent がこの比率より小さい場合は、未使用ページを検索せずに新しいエクステントを割り付けます。

たとえば、120 の OAM ページと未使用データ・ページを 1 つだけ持つ 10GB のテーブルにデータを挿入する場合は、次のようになります。

- page utilization percent が 100 ならば、未使用のデータ・ページを見つけるために 120 の OAM ページをすべてスキャンします。
- page utilization percent が 95 ならば、95 は使用済みページ数と未使用ページ数の合計に対する使用済みページ数の比率より小さいので、Adaptive Server は新しいエクステントをオブジェクトに割り付けます。

page utilization percent の値が小さいと未使用ページが多くなります。page utilization percent の値が大きいと、大規模なテーブルではページの割り付けに時間がかかります。これは、Adaptive Server は新しいエクステントを割り付ける前に未使用ページを見つけるために OAM のスキャンを実行するからです。これによって論理 I/O と物理 I/O が増加します。

ページの割り付けが (特に大量のデータ挿入の場合に) 遅いような場合は、page utilization percent の値を小さくできますが、データを挿入し終わったらリセットしてください。値を小さく設定するとサーバ上のすべてのテーブルに影響を与え、結果としてすべてのテーブルに未使用ページができることになります。

高速バルク・コピーは page utilization percent の設定を無視して、データベース内に使用できるエクステントがなくなるまで、常に新しいエクステントを割り付けます。

partition groups

要約	
デフォルト値	1024
値の範囲	1 ～ 2147483647
ステータス	動的
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	メモリ使用、メタデータ・キャッシュ

partition groups パラメータは、Adaptive Server によって割り付け可能なパーティション・グループの最大数を指定します。パーティション・グループは、Adaptive Server がテーブルの各パーティションへのアクセスを制御するために使用する内部構造体です。パーティション・グループは、アップグレードまたは load database アップグレードで使用され、Adaptive Server 12.5.x 以前のパーティションの分割が解除されます。

このデフォルト値は、最大 1024 のオープン・パーティション・グループと最大 2147483647 のオープン・パーティションを可能にします。実際のパーティションの数は、パーティションのグループ化によりこれよりもやや少なくなることがあります。

partition spinlock ratio

要約	
デフォルト値	10
値の範囲	1 ～ 2147483647
ステータス	動的
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	メモリ使用、メタデータ・キャッシュ

partition spinlock ratio は、Adaptive Server の複数のエンジンが実行されている場合に、1 つの **spinlock** によって保護されるパーティション記述子内のローの数を設定します。

Adaptive Server では、テーブル・パーティションへのアクセスはパーティション記述子を使用して管理されます。各パーティション記述子は、プロセスがそのパーティションにアクセスするときに使用しなければならないパーティション情報 (たとえば、そのパーティションの最終ページ) を保管します。number of open partitions を使用して、パーティション記述子を設定します。

partition spinlock ratio のデフォルト値は、パーティション・キャッシュ 10 個ごとに 1 つのスピンロックを設定します。partition spinlock ratio パラメータの値を減らしても、Adaptive Server のパフォーマンスにはほとんど影響はありません。このデフォルト設定は、ほとんどのサーバに適しています。

『システム管理ガイド 第 2 巻』の「第 5 章 マルチプロセッサ・サーバの管理」を参照してください。

pci memory size

要約	
デフォルト値	64MB
有効な値	0 〜 2147483647
ステータス	動的
表示レベル	中間
必要な役割	システム管理者
設定グループ	ユーザ環境

pci memory size は、PCI (Pluggable Component Interface) メモリ・プールのサイズを設定します。PCI Bridge の下で稼働するすべての JVM プラグインとプラグ可能なコンポーネント アダプタ (PCA: Pluggable Component Adapter) は、1 つの専用 PCI メモリ・プールを共有します。pci memory size がデフォルト値より低い値に設定されても、Adaptive Server はデフォルト値を使用します。

このメモリ・プールは、完全に PCI Bridge と稼働中のプラグ可能コンポーネント専用です。その他すべてのメモリ・プールと同様に、Adaptive Server はこのメモリ・プールを制御します。ただし、他のメモリ・プールとは異なり、PCI のメモリ・プールは PCI Bridge の初期化時に割り付けられ、それ以降に大きくなることはありません。

per object statistics active

要約	
デフォルト値	0 (オフ)
値の範囲	0 (オフ)、1 (オン)
ステータス	動的
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	モニタリング

per object statistic active は、Adaptive Server でオブジェクトごとの統計を収集するかどうかを制御します。

percent database for history

要約	
デフォルト値	20
有効な値	0 ～ 100
ステータス	動的
表示レベル	中間
必要な役割	システム管理者
設定グループ	SQL Server 管理

percent database for history は、js_history テーブルに予約されている sybmgmtdb で使用できる合計領域のパーセンテージを指定します。実行されるジョブが多い場合、または今後のクエリのために実行可能ジョブに関する履歴レコードを保存する必要がある場合は、percent database for history の値を大きくしてください。

percent database for output

要約	
デフォルト値	30
有効な値	0 ～ 100
ステータス	動的
表示レベル	中間
必要な役割	システム管理者
設定グループ	SQL Server 管理

percent database for output は、ジョブ出力に予約されている sybmgmtdb で使用できる合計領域のパーセンテージを指定します。実行されるジョブが多いか、または大量の出力を生成するジョブがあり、その出力をクエリのために保存する必要がある場合は、デフォルト値を増やしてください。

percent history free

要約	
デフォルト値	30
有効な値	0 ～ 100
ステータス	動的
表示レベル	中間
必要な役割	システム管理者
設定グループ	SQL Server 管理

percent history free は、空き領域にしておく必要がある sybmgmtdb 内の予約領域のパーセンテージを指定します。たとえば、デフォルト値が使用される場合、sybmgmtdb の 70% がいっぱいになると、Adaptive Server は最も古い履歴レコードを消去して、新しいレコードを保存するための領域を確保します。

percent output free

要約	
デフォルト値	50
有効な値	0 ～ 100
ステータス	動的
表示レベル	中間
必要な役割	システム管理者
設定グループ	SQL Server 管理

sybmgmtdb 内の領域で、Job Scheduler の出力を格納するために空けておく予約領域をパーセンテージで指定します。たとえば、デフォルト値を使用する場合、sybmgmtdb の 50% がいっぱいになると、Adaptive Server は最も古い履歴レコードを消去して、新しいレコードを保存するための領域を確保します。

performance monitoring option

要約	
デフォルト値	0 (オフ)
値の範囲	0 (オフ)、1 (オン)
ステータス	動的
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	モニタリング

performance monitoring option は、BMC DBXray グラフィカル・パフォーマンス・モニタリングおよび診断ツールのライセンスを有効にします。

permission cache entries

要約	
デフォルト値	15
値の範囲	1 ～ 2147483647
ステータス	動的
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	メモリ使用、ユーザ環境

permission cache entries は、タスク当たりのキャッシュ・プロテクタ数を指定します。その際、各ユーザ接続とワーカー・プロセスのメモリ量が増やされます。

ユーザ・パーミッションに関する情報は、パーミッション・キャッシュに保持されます。Adaptive Server は、パーミッションをチェックするとき、最初にパーミッション・キャッシュを検索します。必要とするパーミッションが見つからない場合は sysprotects テーブルを検索します。このプロセスは、必要とする情報がパーミッション・キャッシュ内に見つかれば、sysprotects を読み込む必要がないので、非常に短時間で検索が終わります。

ただし、Adaptive Server がパーミッション・キャッシュを検索するのはユーザ・パーミッションをチェックするときだけで、パーミッションを付与するときや取り消すときには検索しません。パーミッションの付与または取り消しが行われるときは、パーミッション・キャッシュ全体がフラッシュされます。これは、新しいパーミッションを付与したり、無効にしたりすることによって、既存のパーミッションのタイム・スタンプが古くなるからです。

Adaptive Server 上のユーザが、パーミッションのチェックを必要とする操作を頻繁に実行する場合は、permission cache entries の値を増やすことによってパフォーマンスがわずかに改善されることがあります。この効果は、広範な調整を保証できるほど大きなものではありません。

Adaptive Server 上のユーザがパーミッションの付与や取り消しを頻繁に実行する場合は、permission cache entries の設定値を大きくしないでください。grant コマンドと revoke コマンドを実行するたびにキャッシュがフラッシュされるので、パーミッション・キャッシュのための領域が無駄になります。

plan text pipe active

要約	
デフォルト値	0
値の範囲	0 ～ 1
ステータス	動的
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	モニタリング

plan text pipe active は、Adaptive Server でクエリ・プラン・テキストを収集するかどうかを決定します。plan text pipe active と plan text pipe max messages を両方とも有効にすると、Adaptive Server は各クエリのプラン・テキストを収集します。monSysPlanText を使用すると、すべてのユーザ・タスクのクエリ・プラン・テキストを取得できます。

plan text pipe max messages

要約	
デフォルト値	0
値の範囲	0 ～ 2147483647
ステータス	動的
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	メモリ使用、モニタリング

plan text pipe max messages は、Adaptive Server がエンジンごとに格納するクエリ・プラン・テキストのメッセージ数を決定します。monSQLText テーブル内のメッセージ数の合計は、sql text pipe max messages に実行中のエンジン数を掛け合わせた値になります。

print deadlock information

要約	
デフォルト値	0 (オフ)
有効な値	0 (オフ)、1 (オン)、2 (オン、概要を出力)
ステータス	動的
表示レベル	中間
必要な役割	システム管理者
設定グループ	ロック・マネージャ、SQL Server 管理

print deadlock information は、エラー・ログにデッドロック情報を出力します。

デッドロックが繰り返し発生する場合は、print deadlock information を 1 に設定すると、デッドロックの原因を調べるのに役立つエラー・ログの詳細情報が得られます。ただし、print deadlock information を 1 に設定すると、Adaptive Server のパフォーマンスが低下することがあります。このため、print deadlock information を設定するのは、デッドロックの原因を調べる場合だけにしてください。

使用しているアプリケーションでデッドロックが発生しているかどうかを判断するには、sp_sysmon を使用してください。デッドロックが発生している場合は、print deadlock information パラメータを 1 に設定し、その発生原因の詳細を調べてください。『パフォーマンス＆チューニング・シリーズ：sp_sysmon による Adaptive Server の監視』を参照してください。

値を 2 に設定すると、デッドロック情報の概要をエラー・ログに出力できます (値を 1 に設定すると、詳細情報が出力されます)。次に例を示します。

```
Deadlock Id 34: Process (Familyid 0, Spid 70) was waiting for a 'exclusive page'
lock on page 10858346 of the 'equineline_job' table in database 18 but process
(Familyid 0, Spid 88) already held a 'exclusive page' lock on it.
Deadlock Id 34: Process (Familyid 0, Spid 88) was waiting for a 'exclusive page'
lock on page 11540986 of the 'equineline_job' table in database 18 but process
(Familyid 0, Spid 70) already held a 'update page' lock on it.
```

print recovery information

要約	
デフォルト値	0 (オフ)
有効な値	0 (オフ)、1 (オン)
ステータス	動的
表示レベル	中間
必要な役割	システム管理者
設定グループ	バックアップとリカバリ

print recovery information は、リカバリ中に Adaptive Server のコンソールに表示される情報の内容を指定します (Adaptive Server の起動時およびデータベース・ダンプをロードしたときに、各データベースのリカバリが実行されます)。デフォルト値では、データベース名と、リカバリが進行中であることを知らせるメッセージだけが表示されます。値が 1 の場合、リカバリ中に処理される各トランザクションについての情報が表示されます。これには、そのトランザクションがアボートされたかコミットされたかも含まれます。

procedure cache size

要約	
デフォルト値	7000
値の範囲	7000 ～ 2147483647
ステータス	動的
表示レベル	基本
必要な役割	システム管理者
設定グループ	メモリ使用、SQL Server 管理

プロシージャ・キャッシュのサイズを 2K ページ単位で指定します。Adaptive Server は、ストアド・プロシージャを実行している間はプロシージャ・キャッシュを使用します。キャッシュ内に既にプロシージャのコピーが存在していれば、そのプロシージャをディスクから読み込む必要はありません。Adaptive Server は、ストアド・プロシージャの作成時にクエリをコンパイルするときにも、プロシージャ・キャッシュ内の領域を使用します。

procedure cache size の最適値はアプリケーションごとに異なるので、設定を変更することによって Adaptive Serve のパフォーマンスが向上する可能性があります。たとえば、多数の異なるプロシージャやアドホック・クエリを実行する場合は、アプリケーションがプロシージャ・キャッシュを大量に使用するため、この値を増やす必要がある場合があります。

警告！ procedure cache size の値が小さすぎると、Adaptive Server のパフォーマンスが低下します。

アップグレードするときには、procedure cache size はアップグレード時点のプロシージャ・キャッシュ・サイズに設定されます。

procedure deferred compilation

要約	
デフォルト値	1 (有効)
値の範囲	0 ～ 1
ステータス	動的
表示レベル	
必要な役割	システム管理者
設定グループ	クエリ・チューニング

このパラメータが有効の場合、ストアド・プロシージャ内のローカル変数または一時テーブルを参照する文のコンパイルは延期されます。これは、これらの文の最適化の際に、推定値やマジック・ナンバーではなく実行時値を使用できるようにするためです。

process wait events

要約	
デフォルト値	0 (オフ)
値の範囲	0 (オフ)、1 (オン)
ステータス	動的
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	メモリ使用、モニタリング

process wait events は、Adaptive Server ですべてのタスクの待機イベントごとの統計を収集するかどうかを制御します。monProcessWaits を使用すると、特定のタスクの待機情報を取得できます。

『Transact-SQL ユーザーズ・ガイド』の「第 17 章 ストアド・プロシージャの使用」を参照してください。

prod-consumer overlap factor

要約	
デフォルト値	20
値の範囲	
ステータス	動的
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	

prod-consumer overlap factor は、最適化に影響を与えるパラメータです。Adaptive Server では、group by アルゴリズムが変更されています。並列プランに set statistics I/O を使用することはできません。

quorum heartbeat interval

要約	
デフォルト値	5
有効な値	1 ～ 60
ステータス	動的
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	共有ディスク・クラスタ

quorum heartbeat interval は、クォーラム・ハートビート間の秒数を指定します。quorum heartbeat interval を低い値に設定すると、ハートビート・オーバーヘッドが増加しますが、失われたディスク・リンクの検出が速くなります。その結果、I/O フェンシングを設定したインスタンスまたは SAN リンクを失ったインスタンスがすぐに終了します。quorum heartbeat interval を高い値に設定すると、ハートビートのオーバーヘッドが減少しますが、失われたディスク・リンクの検出が遅くなります。

quorum heartbeat retries

要約	
デフォルト値	2
有効な値	0 ～ 32767
ステータス	動的
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	共有ディスク・クラスタ

read committed with lock

要約	
デフォルト値	0 (オフ)
有効な値	0 (オフ)、1 (オン)
ステータス	動的
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	ロック・マネージャ

read committed with lock は、Adaptive Server がトランザクション独立性レベル 1 (コミット済みを読み込み) を使用している場合の select クエリ実行中に、データオンリーロック・テーブルのローまたはページに対して共有ロックを保持するかどうかを指定します。カーソルについては、読み込み専用として宣言されたカーソルにのみ、read committed with lock が適用されます。

トランザクション独立性レベルが 1 の場合は、全ページロック・テーブルに対して select クエリを実行すると、現在位置のページに対するロックが引き続き保持されます。データオンリーロック・テーブルの更新可能なカーソルも、現在のページまたはローに対するロックを保持します。『パフォーマンス＆チューニング・シリーズ：基本』を参照してください。

recovery interval in minutes

要約	
デフォルト値	5
値の範囲	1 ～ 32767
ステータス	動的
表示レベル	基本
必要な役割	システム管理者
設定グループ	バックアップとリカバリ

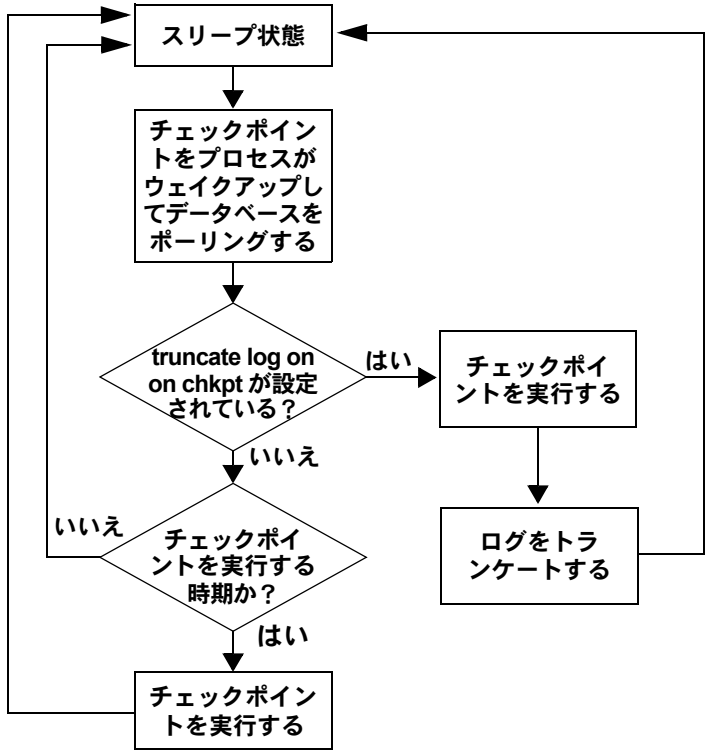
recovery interval in minutes は、システム障害が発生した場合に Adaptive Server がリカバリ・プロシージャを完了するために使用する、データベース当たりの最大時間を分単位で設定します。リカバリ・プロシージャは、チェックポイント・プロセスによって最も古いアクティブ・トランザクションとして記録されているトランザクションから始まる、トランザクションのロールバックまたはロールフォワードを行います。リカバリ・プロセスは recovery interval in minutes の値に応じた作業量を実行します。

Adaptive Server は、トランザクション・ログ内の 6,000 ローに対して 1 分間のリカバリ時間が必要であると見積もります。ただし、ログ・レコードのタイプによっては、リカバリにかかる時間も多少変わります。recovery interval in minutes を 3 に設定した場合は、最後のチェックポイント以降の syslogs のロー数が 18,000 を超えるまでは、変更されたページがチェックポイント・プロセスによってディスクに書き込まれることはありません。

注意　Adaptive Server の障害発生時にアクティブになっていたトランザクションのうち、実行時間が長く、ログの量が非常に少ないトランザクション (たとえば、create index) には、リカバリ間隔の効果はありません。このようなトランザクションの処理を元に戻すには、トランザクションを実行するのと同じぐらいの時間がかかります。あまり時間がかからないようにするには、インデックスの保守操作後に、それぞれのデータベースをダンプしてください。

Adaptive Server は、recovery interval in minutes の設定とそれぞれのデータベースについてのアクティビティの量を使用して、それぞれのデータベースにチェックポイントを実行する時期を決定します。Adaptive Server がデータベースのチェックポイントを実行すると、すべての「ダーティ・ページ」(キャッシュ内にある変更されたデータ・ページ) がディスクに書き込まれます。このとき、「チェックポイント・スパイク」と呼ばれる、短時間で大量の I/O が行われる状態が発生することがあります。チェックポイントの実行時には、この他に、truncate log on chkpt オプションが設定されているデータベースのトランザクション・ログのトランケートなどの管理タスクも実行されます。スリープしているチェックポイント・プロセスは 1 分に約 1 回の割合で「ウェイクアップ」して、truncate log on chkpt の設定をチェックし、リカバリ間隔をチェックしてチェックポイントが必要かどうかを調べます。図 5-4 は、このプロセスで使用されるロジックを示しています。

図 *5-4:* チェックポイント・プロセス



アプリケーションとその使用形態に変更がある場合は、リカバリ間隔を変更します。たとえば、Adaptive Server の更新アクティビティが増加した場合は、リカバリ間隔を短くします。リカバリ間隔を短くするとチェックポイントの頻度が上がり、チェックポイント・スパイクは小さくなりますが頻繁に発生するようになり、システムの処理速度が多少遅くなります。一方、リカバリ間隔が長すぎると、リカバリ時間が長くなりすぎる可能性があります。チェックポイントによって生じるスパイク数を減らすには、housekeeper freewrite percent パラメータを再設定します。「housekeeper free write percent」(137 ページ) を参照してください。パフォーマンスへの recovery interval in minutes の影響については、『パフォーマンス＆チューニング・シリーズ：基本』の「第 5 章 メモリの使い方とパフォーマンス」を参照してください。

特定のリカバリ間隔がシステムにどのように影響するかを判断するには sp_sysmon を使用してください。『パフォーマンス＆チューニング・シリーズ：sp_sysmon による Adaptive Server の監視』を参照してください。

remote server pre-read packets

要約	
デフォルト値	3
値の範囲	3 ～ 255
ステータス	静的
表示レベル	中間
必要な役割	システム管理者
設定グループ	メモリ使用、ネットワーク通信

remote server pre-read packets は、リモート・サーバとの接続中にサイト・ハンドラが「先読み」するパケットの数を決定します。

必要な接続数を少なくするために、2 つのサーバ間の通信を 1 つのサイト・ハンドラで管理します。サイト・ハンドラは、受け取り側のプロセスがデータ・パケットを受け取る準備ができる前に、それぞれのユーザ・プロセスのためのデータ・パケットを先読みして追跡することができます。

remote server pre-read packets のデフォルト値は、ほとんどのサーバに適した値です。この値を大きくするとメモリの使用量が増え、小さくするとサーバ間のネットワーク通信が遅くなります。「第 15 章 リモート・サーバの管理」を参照してください。

restricted decrypt permission

要約	
デフォルト値	0
値の範囲	0 (オフ)、1 (オン)
ステータス	動的
表示レベル	基本
必要な役割	システム・セキュリティ担当者
設定グループ	セキュリティ関連

restricted decrypt permission は、すべてのデータベースにおける制限のある decrypt パーミッションを有効または無効にします。このパラメータを設定するには、sso_role が必要です。

restricted decrypt permission を 0 (オフ) に設定すると、暗号化カラムに対する decrypt パーミッションの機能は 15.0.2 より前のバージョンと同じになります。

- テーブル所有者または SSO が明示的に decrypt パーミッションを付与します。ただし、decrypt パーミッションに対する with grant option がサポートされています。

- decrypt パーミッションは、テーブル所有者と SSO だけでなく、所有権の連鎖によってどのユーザにも暗黙的に付与されます。たとえば、ユーザ Fred が proc1 ストアド・プロシージャを所有しており、これにより、暗号化カラム fred.table1.col1 からデータが選択されるとします。Fred が proc1 の exec パーミッションを Harry に付与した場合、Harry は fred.table1.col1 に対する暗黙的な decrypt パーミッションを取得します。
- decrypt パーミッションは、alter table decryp には必要ありません。これは、テーブル所有者が、暗号化カラムに対する暗黙的な decrypt パーミッションを取得しているためです。

restricted decrypt permission を 1 (オン) に設定した場合

- SSO にのみ decrypt パーミッションが暗黙的に付与されます。
- SSO は、with grant option パラメータを使用して decrypt パーミッションを付与できます。これにより、SSO はシステム内のどのユーザが decrypt パーミッションを付与できるかを決定できます。たとえば、user1 が user3.user3_tab に対する decrypt permission を付与できるようにする場合、SSO は次のコマンドを発行します。

  ```
  grant decrypt on user3.user3_tab to user1
  with grant option
  ```

  システム暗号化パスワードを使用する場合は、データのプライバシーを保護するために、decrypt パーミッションを DBO に付与しないことをおすすめします。ユーザ・パスワードを介したキーへのアクセスにより、DBO やその他のグループは、キーのパスワードを取得しない限り、データにアクセスできません。しかし、暗号化されたデータを表示できるユーザを DBO が決定するようにすると便利です。ユーザが指定したパスワードでキーとデータを保護しない場合は、SSO だけが decrypt パーミッションを付与する責任を持つようにしてください。
- テーブル所有者であっても、暗黙的な decrypt パーミッションは付与されません。つまり、暗号化カラムを含むテーブルを作成しても、明示的に付与されない限り、これらのカラムに対する decrypt パーミッションを取得しません。
- 所有権の連鎖によって、decrypt パーミッションが暗黙的に付与されることはありません。たとえば、Fred が proc1 ストアド・プロシージャを所有し、これにより、暗号化カラム fred.table1.col1 からデータが選択されるとします。Fred が proc1 の exec パーミッションを Harry に付与した場合、Harry がデータを表示するには、fred.table1.col1 に対する明示的な decrypt パーミッションも取得する必要があります。
- エイリアス・ユーザは、エイリアスが設定されているユーザのパーミッションを代用できます。同様に、いずれかのデータベース内の DBO にエイリアスが暗黙的に設定された、sa_role を持つユーザは、DBO に明示的に付与された decrypt パーミッションを継承します。

- テーブル所有者は所有するテーブルに対する暗黙的な decrypt パーミッションを持たないため、alter table decrypt 文には decrypt パーミッションが必要です。

restricted decrypt permission を 0 から 1 に変更すると、暗黙的 decrypt パーミッションを使用する、現在実行中の文が完了します。SSO が、必要なカラムに対する decrypt パーミッションをユーザに付与するまで、暗黙的 decrypt パーミッションを使用する後続の文は正常に実行されず、次のエラー・メッセージを返します。

```
Msg 10330 "DECRYPT permission denied on object object_name, database
database_name, owner owner_name."
```

restricted decrypt permission を 1 から 0 に変更すると、明示的な付与を反映するローは sysprotects システム・テーブルに保持されます。ただし、Adaptive Server は、decrypt パーミッションを暗黙的に付与できるかどうかを確認するために sysprotects をチェックしないため、これらのローは暗黙的に付与された decrypt パーミッションには影響を与えません。システムが再設定される前に明示的な decrypt パーミッションが付与されたか取り消され、暗黙的な decrypt パーミッションを現在持っているユーザにのみ、sp_helprotect は誤解を招く情報を表示します。

システムの一貫性を維持するため、ユーザに付与した明示的な decrypt パーミッションを取り消してから、restricted decrypt permission の有効または無効を切り替え、システムの一貫性を維持することをおすすめします。

decrypt パーミッションの詳細については、『暗号化カラム・ユーザーズ・ガイド』を参照してください。

row lock promotion HWM

要約	
デフォルト値	200
値の範囲	2 ～ 2147483647
ステータス	動的
表示レベル	中間
必要な役割	システム管理者
設定グループ	ロック・マネージャ、SQL Server 管理

row lock promotion HWM (high-water mark) は、row lock promotion LWM (low-water mark) と row lock promotion PCT とともに使用します。このパラメータは、テーブルまたはインデックスの 1 回のスキャン・セッション中に許容されるロー・ロックの最大数を指定します。この数に達すると、Adaptive Server はロー・ロックからテーブル・ロックへの拡大を試みます。

スキャン・セッション中に取得したロックの数が row lock promotion HWM を超えると、Adaptive Server はテーブル・ロックの取得を試みます。lock promotion HWM の値を、number of locks の値より大きくすることはできません。

『パフォーマンス&チューニング・シリーズ：ロックと同時実行制御』の「第 2 章 ロックの設定とチューニング」を参照してください。

row lock promotion HWM のデフォルト値は、ほとんどのアプリケーションに適した値です。テーブルのロックを回避するには、row lock promotion HWM の値を大きくします。たとえば、数千ものローがあるテーブルのうち 500 ローに対して定期的な更新が行われることがわかっている場合は、row lock promotion HWM を 500 前後に設定すれば、このようなテーブルの同時実行性を高めることができます。

ロー・ロック・プロモーションはオブジェクト単位のレベルでも設定できます。『リファレンス・マニュアル：プロシージャ』の「sp_setpglockpromote」を参照してください。

row lock promotion LWM

要約	
デフォルト値	200
値の範囲	2 ～ row lock promotion HWM の値
ステータス	動的
表示レベル	中間
必要な役割	システム管理者
設定グループ	ロック・マネージャ、SQL Server 管理

row lock promotion LWM (low-water mark) は、row lock promotion HWM (high-water mark) と row lock promotion PCT (percentage) とともに使用します。このパラメータは、テーブルまたはインデックスの 1 回のスキャン・セッション中に許容されるロー・ロックの数を指定します。この数に達すると、Adaptive Server はロー・ロックからテーブル・ロックへの拡大を試みます。

row lock promotion LWM で設定されたロック数に達するまでは、Adaptive Server はオブジェクトに対するテーブル・ロックの取得を試みません。row lock promotion LWM パラメータには、row lock promotion HWM 以下の値を設定してください。

row lock promotion LWM のデフォルト値は、ほとんどのアプリケーションに適した値です。Adaptive Server がロックを使い果たす状況が繰り返し発生する場合は、number of locks を増やしてください。

詳細については、『パフォーマンス&チューニング・シリーズ：ロックと同時実行制御』を参照してください。

ロック・プロモーションは、オブジェクト単位のレベルで設定することもできます。『リファレンス・マニュアル：プロシージャ』の「sp_setpglockpromote」を参照してください。

row lock promotion PCT

要約	
デフォルト値	100
値の範囲	1 ～ 100
ステータス	動的
表示レベル	中間
必要な役割	システム管理者
設定グループ	ロック・マネージャ、SQL Server 管理

オブジェクトに対して保持されているロック数が row lock promotion LWM (low-water mark) と row lock promotion HWM (high-water mark) の間の値である場合に、row lock promotion PCT で設定されたロー・ロックのパーセンテージ (テーブル内のロー数に基づく) を超えると、Adaptive Server はテーブル・ロックの取得を試みます。

row lock promotion PCT のデフォルト値は、ほとんどのアプリケーションに適した値です。

ロック・プロモーションの制限値のセットアップ方法の詳細については、『パフォーマンス＆チューニング・シリーズ：ロックと同時実行制御』の「第 2 章 ロックの設定とチューニング」を参照してください。

ロー・ロック・プロモーションはオブジェクト単位のレベルでも設定できます。『リファレンス・マニュアル：プロシージャ』の「sp_sterowlockpromote」を参照してください。

rtm thread idle wait period

要約	
デフォルト値	600
値の範囲	600 ～ 4026531839
ステータス	動的
表示レベル	中間
必要な役割	システム管理者
設定グループ	SQL Server 管理

rtm thread idle wait period は、Adaptive Server が使用するネイティブ・スレッドが無処理のときの待ち時間の長さを秒単位で定義します。

runnable process search count

要約	
デフォルト値	2000 (クラスタ・エディションではデフォルト値は 3)
値の範囲	0 ～ 2147483647
ステータス	動的
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	SQL Server 管理

runnable process search count パラメータは、実行可能なタスクがあるかどうかを見つけるためにエンジンがループする回数を指定します。この数に達すると、CPU はオペレーティング・システムに解放されます。

Adaptive Server のエンジンは、タスクが完了したときや、そのエンジンでの割り当てられた時間を過ぎたときに、実行キューに実行可能なタスクがあるかどうかを調べます。場合によっては、実行キューにタスクがないこともあります。この場合、エンジンは、CPU をオペレーティング・システムに解放することも、実行するタスクを探し続けることもできます。runnable process search count を大きな値に設定すると、エンジンがより多くの回数ループするようになり、結果として長い時間 CPU を保持することになります。runnable process search count を小さな値に設定すると、エンジンは短時間で CPU を解放します。

マシンがユニプロセッサであり I/O の実行をヘルパ・スレッドに依存している場合は、runnable process search を設定してネットワーク I/O、ディスク I/O、または他のオペレーティング・システム・タスクを実行すると、パフォーマンスが良くなることがあります。バルク・コピー処理などのクライアントの処理を、ヘルパ・スレッドを使用する 1 つの CPU サーバと同じマシン上で実行している場合は、サーバとクライアントの両方を CPU にアクセスできるようにする必要があるかもしれません。

注意 パフォーマンス上の問題がある場合は、runnable process search count を 3 に設定してください。

ヘルパ・スレッドを使用しないユニプロセッサのマシンで Adaptive Server を稼動させる場合や、マルチプロセッサ・マシンの場合は、デフォルト値で十分なパフォーマンスが得られます。

runnable process search count の値を 3 に設定すると、クラスタ・エディションでシステム CPU が同じマシン上で稼働している他のプロセスとさらに共有されるようになります。ただし、runnable process search count が 3 で、しかも Adaptive Server がスタンドアロン・プロセスとして稼働している場合、サーバ応答時間が長くなる可能性があります。この場合には、runnable process search count を 2000 に再設定します。

runnable process search count パラメータが Adaptive Server による CPU サイクルの使用、エンジンからオペレーティング・システムへの CPU の解放、ブロッキング・ネットワーク・チェックにどのように影響するかを調べるには、sp_sysmon を使用してください。『パフォーマンス＆チューニング・シリーズ：sp_sysmon による Adaptive Server の監視』を参照してください。

sampling percent

要約	
デフォルト値	0
値の範囲	0 ～ 100 パーセント
ステータス	動的
表示レベル	包括
必要な役割	システム管理者またはデータベース管理者
設定グループ	クエリ・チューニング

sampling percent は、サンプリング率を表す数値です。5% の場合は 5、10% の場合は 10 のように指定します。

I/O の競合とリソースを減らすには、サンプリング方式を使用して update statistics を実行します。これにより、メンテナンス時間が少なく、データ・セットが大きい場合の I/O と時間を削減できます。常時使用され、トランケートおよび再移植される大規模なデータ・セットまたはテーブルを更新する場合は、統計的サンプリングを行うことによって、時間と I/O サイズを削減できます。

結果が十分に正確とは限らないので、サンプリングには注意が必要です。ヒストグラム値の変化と I/O の節減のバランスをとってください。

データ・セットのサンプリングは完全に正確とは言えませんが、通常はヒストグラムも密度値も許容範囲内に収まります。

サンプリングを使用するかどうかを判断するときは、データ・セットのサイズ、作業時間の制約、生成されるヒストグラムが必要な程度に正確であるかを考慮に入れてください。

サンプリングで使用するパーセンテージは要件に応じて異なります。特定のデータ・セットについての情報を最も正確に反映した結果が得られるまで、さまざまなパーセンテージをテストしてください。

統計値は、システム・テーブル systabstats と sysstatistics に格納されます。

secure default login

要約	
デフォルト値	0
値の範囲	0 (これに続くパラメータでデフォルト・ログイン名を指定)
ステータス	動的
表示レベル	中間
必要な役割	システム・セキュリティ担当者
設定グループ	セキュリティ関連

secure default login パラメータは、事前に認証されているが master..syslogins にログインが登録されていないすべてのユーザに対するデフォルト・ログインを指定します。

セキュア・デフォルト・ログインは、次の構文を使用して設定します。

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

構文の説明は、次のとおりです。

- secure default login はパラメータの名前です。
- 0 は必須パラメータです。sp_configure の 2 番目のパラメータは数値にする必要があるためです。
- *default_login_name* は、Adaptive Server のユーザとしては登録されていないが、セキュリティ・メカニズムによって既に認証されているユーザ用のデフォルト・ログイン名です。ログイン名は、master..syslogins に存在する有効なログインでなければなりません。

select on syscomments.text

要約	
デフォルト値	1 (オン)
値の範囲	0 (オフ)、1 (オン)
ステータス	動的
表示レベル	包括
必要な役割	システム・セキュリティ担当者
設定グループ	セキュリティ関連

select on syscomments.text は、syscomments テーブルの text カラムに対する select パーミッションを制限することにより、データベース・オブジェクト・テキストの保護を有効にします。デフォルト値は、select パーミッションを “public” に設定します。値を 0 に設定すると、select パーミッションをオブジェクトの所有者とシステム管理者だけに制限します。

send doneinproc tokens

要約	
デフォルト値	1 (オン)
値の範囲	0 (オフ)、1 (オン)
ステータス	動的
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	ネットワーク通信

send doneinproc tokens は、Adaptive Server による doneinproc パケット (ストアド・プロシージャで実行される各 select 文の後に送信される TDS メッセージ) の送信を有効または無効にします。send doneinproc tokens は、dbcc tune ‘doneinproc’ およびトレース・フラグ 292 に代わるものです。オプションに変更があれば、現在実行中のクエリがすぐに検出します。

多くの場合、send doneinproc tokens を 1 に設定すると無難です。ただし、一部のストアド・プロシージャは CT-Lib から非同期コマンドを使用して実行され、値 0 を使用すると、一部の CT-Lib アプリケーションで状態マシン・エラーが発生することがあります。

session migration timeout

要約	
デフォルト値	600
有効な値	0 ～ 32767
ステータス	動的
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	共有ディスク・クラスタ

session migration timeout は、クライアントがターゲット・インスタンスに接続してマイグレーションを完了するために使用できる時間を指定します。クライアントが割り付けられた時間内にターゲット・インスタンスにマイグレートしない場合、 Adaptive Server は接続できなくなります。

session tempdb log cache size

要約	
デフォルト値	論理ページ・サイズ
値の範囲	2147483647 までの論理ページ・サイズ
ステータス	静的
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	ユーザ環境

session tempdb log cache size は、ユーザ・ログ・キャッシュ (ULC: User Log Cache) のサイズを設定し、必要なフラッシュの回数を指定できるようにします。

shared memory starting address

要約	
デフォルト値	0
値の範囲	プラットフォーム固有
ステータス	静的
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	物理メモリ

shared memory starting address は、Adaptive Server の共有メモリ領域の開始仮想アドレスを決定します。

shared memory starting address を設定し直す必要はほとんどありません。設定を検討する場合は、Sybase の保守契約を結んでいるサポート・センタに事前にご相談ください。

number of worker processes、max parallel degree、max scan parallel degree は、並列クエリ処理をサーバ・レベルで制御します。set コマンドで parallel_degree オプション、process_limit_action オプション、scan_parallel_degree オプションを使用すれば、並列処理の最適化をセッション・レベルで制限できます。また、select コマンドで parallel キーワードを使用すれば、そのクエリの並列処理の最適化を制限できます。

size of auto identity column

要約	
デフォルト値	10
値の範囲	1 ～ 38
ステータス	動的
表示レベル	中間
必要な役割	システム管理者
設定グループ	SQL Server 管理

size of auto identity column パラメータは、sp_dboption auto identity オプションと unique auto_identity index オプションで自動的に作成される IDENTITY カラムの精度を設定します。

IDENTITY カラムに挿入できる最大値は、$10^{precision}$ -1 です。IDENTITY カラムが最大値に達すると、後続の insert 文はすべてエラーとなり、現在のトランザクションはアボートします。

IDENTITY カラムの最大値に達した場合に最大値を大きくするには、alter table コマンドを使用して修正オペレーションを実行します。実行例については、『Transact-SQL ユーザーズ・ガイド』を参照してください。

また、create table コマンドを使用して元のテーブルと同じテーブルを作成し、このときに IDENTITY カラムの精度に大きい値を指定する方法もあります。新しいテーブルを作成したら、insert コマンドか bcp を使用して古いテーブルから新しいテーブルにデータをコピーしてください。

size of global fixed heap

要約	
デフォルト値	150 ページ (32 ビット・バージョン)
	300 ページ (64 ビット・バージョン)
最小値	10 ページ (32 ビット・バージョン)
	20 ページ (64 ビット・バージョン)
ステータス	動的
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	Java サービス、メモリ使用

size of global fixed heap は、内部データ構造などに使用するメモリ領域を指定します。

size of the global fixed heap を変更する場合には、論理メモリの合計サイズも同じ量だけ変更してください。

size of process object heap

要約	
デフォルト値	1500 ページ (32 ビット・バージョン)
	3000 ページ (64 ビット・バージョン)
最小値	45 ページ (32 ビット・バージョン)
	90 ページ (64 ビット・バージョン)
ステータス	動的
表示レベル	基本
必要な役割	システム管理者
設定グループ	Java サービス、メモリ使用

size of process object fixed heap は、Java VM を使用するすべてのプロセスで使用できるメモリ領域の合計サイズを指定します。

size of process object fixed heap を変更する場合は、total logical memory も同じ量だけ変更してください。

size of shared class heap

要約	
デフォルト値	1536 ページ (32 ビット・バージョン)
	3072 ページ (64 ビット・バージョン)
最小値	650 ページ (32 ビット・バージョン)
	1300 ページ (64 ビット・バージョン)
ステータス	動的
表示レベル	基本
必要な役割	システム管理者
設定グループ	Java サービス、メモリ使用

size of shared class heap は、Java VM に呼び出されるすべての Java クラスの共有メモリ領域を指定します。Adaptive Server は、ユーザ定義の Java クラスとシステムで提供される Java クラス用の共有クラス・ヒープをサーバ全体で管理します。

size of shared class heap を変更する場合は、total logical memory も同じ量だけ変更してください。

size of unilib cache

要約	
デフォルト値	0
値の範囲	0 ～ 2147483647
ステータス	動的
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	メモリ使用、Unicode

size of unilib cache には、最小オーバヘッド・サイズとは別に使用されるメモリの量 (バイト単位) を 1K 単位に切り上げた値を指定します。最大の Unilib 変換テーブルと最大の Unilib ソート・テーブル全体を一度にロードするのに十分な大きさになるようにします。アジア言語を使用する環境では、Unicode ベースの変換を利用する文字セットを 1 つ追加するたびに、size of unilib cache を 100K 増やします。

sproc optimize timeout limit

要約	
デフォルト値	40
値の範囲	0 ～ 4000
ステータス	動的
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	クエリ・チューニング

sproc optimize timeout limit は、Adaptive Server が保存されたプロシージャを最適化するために費やすことができる時間を、予想実行時間に対する割合として指定します。

SQL batch capture

要約	
デフォルト値	0 (オフ)
値の範囲	0 (オフ)、1 (オン)
ステータス	動的
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	モニタリング

SQL batch capture は、Adaptive Server で SQL テキストを収集するかどうかを制御します。SQL batch capture と max SQL text monitored の両方が有効の場合、Adaptive Server は各ユーザ・タスクのバッチごとに SQL テキストを収集します。

SQL Perfmon Integration (Windows のみ)

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

SQL Perfmon Integration パラメータは、Windows パフォーマンス・モニタによる Adaptive Server の統計のモニタリングを行うかどうかを設定します。

モニタの統合をサポートするには、Adaptive Server を Windows サービスとして登録する必要があります。次の場合に自動的にこの状態になります。

- [Sybase for the Windows] プログラム・グループの [Services Manager] を使用して Adaptive Server を起動した場合
- [コントロール パネル] で [サービス] オプションを使用した場合
- Adaptive Server を自動サービスとして起動するように Windows を設定している場合

Adaptive Server のモニタ可能なカウンタについては、『設定ガイド Windows 版』を参照してください。

sql server clock tick length

要約	
デフォルト値	プラットフォーム固有
値の範囲	プラットフォーム固有の最小値 ～ 1000000、ただしデフォルト値の倍数
ステータス	静的
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	SQL Server 管理

sql server clock tick length は、サーバのクロック・チックの長さをマイクロ秒単位で指定します。デフォルト値と最小値は、どちらもプラットフォーム固有のものです。値は *n* の整数倍に切り上げられます (*n* はプラットフォーム固有のクロック・チックのデフォルト値)。sql server clock tick length の現在の値は、sp_helpconfig または sp_configure を使用して確認できます。

混合使用のアプリケーションにおいて、CPU バウンド・タスクがある場合は、sql server clock tick length の値を小さくして、次の目的を果たします。

- I/O バウンド・タスクを実行しやすくする － これには 20,000 が妥当な値です。クロック・チックの長さを短くすることは、CPU バウンド・タスクがエンジンに割り当てられた時間を超える頻度が増え、これによって他のタスクが CPU を多く使えることを意味します。
- 応答時間をわずかに長くする － Adaptive Server はサービス・タスクをクロック・チックごとに 1 回実行します。クロック・チックの長さを短くすることは、単位時間当たりのサービス・タスクの実行回数が増えることを意味します。

sql server clock tick length の値を増やすと、コンテキストの切り替えと切り替えの間の実行時間が長くなり、CPU バウンド・タスクに有利になります。CPU バウンドの性質を持つアプリケーションにとっては、最大値の 1,000,000 が妥当な値です。ただし、I/O バウンド・タスクにとっては結果的に不利になります。これは cpu grace time (「cpu grace time」(97 ページ) を参照) と time slice (「time slice」(246 ページ) を参照) を調整することである程度は緩和できます。

注意　sql server clock tick length の値を変更すると、Adaptive Server のパフォーマンスに大きな影響を与える可能性があります。この値の設定を変更する場合は、Sybase 製品の保守契約を結んでいるサポート・センタに事前にご相談ください。

sql text pipe active

要約	
デフォルト値	0
値の範囲	0 ～ 1
ステータス	動的
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	モニタリング

sql text pipe active は、Adaptive Server で SQL テキストを収集するかどうかを制御します。このオプションを有効にし、sql text pipe max messages を設定すると、Adaptive Server は各クエリの SQL テキストを収集します。monSysSQLText を使用すると、すべてのユーザ・タスクの SQL テキストを取得できます。

sql text pipe max messages

要約	
デフォルト値	0
値の範囲	0 ～ 2147483647
ステータス	動的
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	メモリ使用、モニタリング

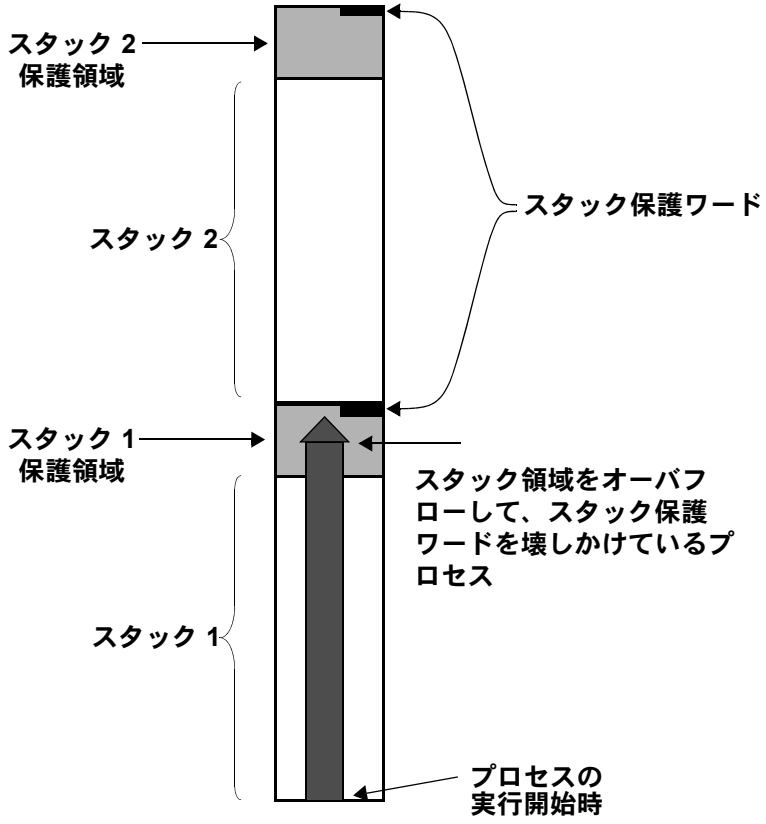
sql text pipe max messages は、Adaptive Server がエンジンごとに格納する SQL テキストのメッセージ数を指定します。monSQLText テーブル内のメッセージ数の合計は、sql text pipe max messages に実行中のエンジン数を掛け合わせた値になります。

stack guard size

要約	
デフォルト値	4096
値の範囲	0 ～ 2147483647
ステータス	静的
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	メモリ使用、ユーザ環境

stack guard size は、スタック保護領域のサイズをバイト単位で設定します。「スタック保護領域」とは、各スタックの終わりにある、サイズの設定が可能なオーバフロー・スタックです。Adaptive Server は、起動時にユーザ接続とワーカー・プロセスのそれぞれに 1 つのスタックを割り付けます。これらのスタックは、それぞれのスタックの終わりに保護領域を伴って同じメモリ領域に連続して配置されます。各スタック保護領域の終わりに、既知のパターンを持つ 4 バイトの構造体である「保護ワード」があります。図 5-5 は、プロセスがスタックの保護ワードを破壊するおそれがあることを示しています。

図 *5-5:* スタック保護ワードを壊しかけているプロセス



Adaptive Server は、ユーザ接続のスタック・ポインタが、そのユーザ接続に関連付けられているスタックのスタック保護領域に侵入したかどうかを定期的にチェックします。侵入している場合は、トランザクションをアボートし、そのトランザクションを生成したアプリケーションに制御を戻して、次のエラー・メッセージ 3626 を生成します。

```
The transaction was aborted because it used too much
stack space. Either use sp_configure to increase the
stack size, or break the query into smaller pieces.
spid: %d, suid: %d, hostname: %.*s, application name:
%.*s
```

また、Adaptive Server は定期的に保護ワードのパターンをチェックして、変更されていないかどうか調べます。変更されている場合は、プロセスがスタック境界をオーバフローしたことを示します。このとき、Adaptive Server はエラー・ログに次のメッセージを出力し、停止します。

```
kernel: *** Stack overflow detected:  limit:  0x%lx sp:  0x%lx
kernel: *** Stack Guardword corrupted
kernel: *** Stack corrupted, server aborting
```

最初のメッセージの “limit” はスタック保護領域の終わりのアドレスで、“sp” はスタック・ポインタの現在の値です。

さらに、Adaptive Server は定期的にスタック・ポインタをチェックして、ポインタのプロセスのスタックとスタック保護領域のどちらからも完全に外に出てしまっているかどうかを調べます。外にある場合、Adaptive Server は保護ワードが壊されていなくても停止します。このとき、Adaptive Server はエラー・ログに次のメッセージを出力します。

```
kernel:  *** Stack overflow detected:  limit:  0x%lx sp:  0x%lx
kernel: *** Stack corrupted, server aborting
```

stack guard size のデフォルト値は、ほとんどすべてのアプリケーションに適した値です。しかし、スタック保護ワードの破壊またはスタック・オーバフローのいずれかによってサーバが停止した場合は、stack guard size を 2K 分増やしてください。設定されたユーザ接続とワーカー・プロセスのそれぞれにスタック保護領域が割り当てられるので、stack guard size を増やすと、その値に設定済みのユーザ接続とワーカー・プロセスの数を掛けた分のメモリを使用することになります。

スタックのオーバフローの問題を避けるには、stack guard size を増やすのではなく、stack size を増やすことを検討してください (「stack size」(236 ページ) を参照してください)。スタック保護領域はオーバフローに備えた領域であって、通常のスタックの拡張のためのものではありません。

Adaptive Server は、stack size パラメータと stack guard size パラメータの値を加算して、各タスクのスタック領域を割り付けます。stack guard size は、2K の倍数となるように設定してください。2K の倍数でない値を指定した場合は、2K の倍数となるように sp_configure 検証ルーチンによって切り上げられます。

stack size

要約	
デフォルト値	プラットフォーム固有
値の範囲	プラットフォーム固有の最小値 ～ 2147483647
ステータス	静的
表示レベル	基本
必要な役割	システム管理者
設定グループ	ユーザ環境

stack size パラメータは、Adaptive Server 上の各ユーザ・プロセスが使用する実行スタックのサイズをバイト単位で指定します。使用するプラットフォームの stack size パラメータの値を確認するには、sp_helpconfig または sp_configure を使用してください。stack size は、2K の倍数となるように設定してください。2K の倍数でない値を指定した場合は、2K の倍数となるように sp_configure 検証ルーチンによって切り上げられます。

「実行スタック」とは、ユーザ・プロセスがプロセス・コンテキストを記録し、ローカル・データを保管するための Adaptive Server メモリの領域です。

クエリによっては、スタック・オーバフローの原因になるものがあります。たとえば、クエリの where 句が極端に長い場合や、select リストが長い場合、ストアド・プロシージャのネストが深い場合、holdlock で複数の select と update を実行する場合などがこれに当たります。スタック・オーバフローが発生すると、Adaptive Server はエラー・メッセージを出力してトランザクションをロールバックします。具体的なエラー・メッセージについては、「stack guard size」(234 ページ) および『トラブルシューティング＆エラー・メッセージ・ガイド』を参照してください。

スタック・オーバフローに対処するための 2 つのオプションは、大きなクエリを小さなクエリに分割するか、stack size を増やすことです。stack size を変更すると、設定されているユーザ接続とワーカー・プロセスそれぞれに必要なメモリ量にも影響します。「total logical memory」(247 ページ) を参照してください。

クエリのサイズが実行スタックのサイズを超えるような場合は、一連の小さなクエリに書き換えます。これは、そのようなクエリの数が少ないか、たまにしか実行しないものである場合に特に当てはまります。

クエリを実際に実行せずにクエリが必要とするスタック領域の量を調べる方法はありません。各ユーザ接続とワーカー・プロセスのスタック領域は、起動時に事前に割り付けられます。

したがって、stack size の適切な値を決定するには、実験が必要です。stack size のデフォルト値を使用して、最も大きく複雑なクエリをテストしてください。エラー・メッセージを生成することなく実行できれば、おそらくデフォルト値で十分です。エラー・メッセージが生成される場合は、stack size を少し (2K) だけ増やしてみます。クエリを再実行し、追加した量で十分かどうかを調べます。十分でない場合は、エラー・メッセージを生成しないで実行できるようになるまで stack size を増やし続けます。

CIS を使用する場合、またはデータベースで Java を有効にし、JDBC を呼び出すメソッドを使用する場合は、このデフォルト値を 1.5 倍に大きくすることをおすすめします。JDBC または CIS を使用しない場合は、通常は標準のデフォルト値で十分です。

start mail session (Windows のみ)

要約	
デフォルト値	0 (オフ)
有効な値	0 (オフ)、1 (オン)
ステータス	動的
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	拡張ストアド・プロシージャ

start mail session は、Adaptive Server 起動時に Adaptive Server のメール・セッションを自動的に開始するかどうかを指定します。

1 を指定すると、次回の Adaptive Server の起動時にメール・セッションを開始するように Adaptive Server が設定されます。0 を指定すると、次回の Adaptive Server の再起動時にメール・セッションを開始しないように Adaptive Server が設定されます。

start mail session が 0 の場合に Adaptive Server のメール・セッションを明示的に開始するには、xp_startmail システム ESP を使用します。

start mail session を設定する前に、Windows システムに Adaptive Server 用のメールボックスとメール・プロファイルを作成する必要があります。次に、Sybmail 用の Adaptive Server アカウントを作成します。『設定ガイド Windows 版』を参照してください。

start xp server during boot

要約	
デフォルト値	0 (オフ)
値の範囲	0 (オフ)、1 (オン)
ステータス	静的
表示レベル	
必要な役割	
設定グループ	拡張ストアド・プロシージャ

start xp server during boot は、Adaptive Server の起動時に XP Server を起動するかどうかを指定します。

1 に設定されると、Adaptive Server の起動時に XP Server も起動します。start xp server during boot が 0 に設定されると、xp_cmdshell が実行されるまでは XP Server は起動しません。

startup delay

要約	
デフォルト値	0 (オフ)
値の範囲	0 (オフ)、1 (オン)
ステータス	動的
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	クエリ・チューニング

startup delay は、サーバの起動時のどの段階で RepAgent を起動するかを制御します。デフォルトでは、RepAgent は Adaptive Server と同時に起動します。Adaptive Server は、待機時間を示すメッセージをエラー・ログに書き込みます。

statement cache size

要約	
デフォルト値	0
有効な値	キャッシュのサイズ (2K ページ単位)
ステータス	動的
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	メモリ使用、SQL Server 管理

statement cache size を使用すると、プロシージャ・キャッシュ・メモリのサーバ割り付けが増え、プロシージャ・キャッシュ・プールのうち文のキャッシュに使用されるメモリ量が制限されます。

注意　ステートメント・キャッシュを有効にする場合は、set chained on/off をバッチ内に設定してください。

キャッシュされた文はライトウェイト・ストアド・プロシージャに変換されるため、文のキャッシュではオープンしているオブジェクト記述子がさらに必要になります。

statement pipe active

要約	
デフォルト値	0 (オフ)
値の範囲	0 (オフ)、1 (オン)
ステータス	動的
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	モニタリング

statement pipe active は、Adaptive Server で文レベルの統計を収集するかどうかを制御します。statement pipe active と statement pipe max messages の両方を有効にすると、Adaptive Server は各クエリの文の統計を収集します。monSysStatement を使用すると、実行されたすべての文に関する統計を取得できます。

statement pipe max messages

要約	
デフォルト値	0
値の範囲	0 ～ 2147483647
ステータス	動的
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	メモリ使用、モニタリング

statement pipe max messages は、Adaptive Server がエンジンごとに格納する文の統計のメッセージ数を決定します。monSQLText テーブル内のメッセージ数の合計は、sql text pipe max messages に実行中のエンジン数を掛け合わせた値になります。

statement statistics active

要約	
デフォルト値	0 (オフ)
値の範囲	0 (オフ)、1 (オン)
ステータス	動的
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	モニタリング

statement statistic active は、Adaptive Server でモニタリング・テーブルの文レベルの統計を収集するかどうかを制御します。monProcessStatement を使用すると、特定のタスクの文の統計を取得できます。

strict dtm enforcement

要約	
デフォルト値	1 (オン)
有効な値	0 (オフ)、1 (オン)
ステータス	静的
表示レベル	10
必要な役割	システム管理者
設定グループ	DTM 管理

strict dtm enforcement は、Adaptive Server トランザクション・コーディネーション・サービスが分散トランザクションの ACID プロパティを厳密に適用するかどうかを決定します。

Adaptive Server がトランザクションを送信し、コーディネートする相手が、トランザクション・コーディネーションをサポートする他の Adaptive Server だけである環境では、strict dtm enforcement をオンに設定します。トランザクション・コーディネーション・サービスをサポートしないサーバにあるデータをトランザクションが更新しようとすると、Adaptive Server はそのトランザクションをアボートします。

異機種間環境では、トランザクション・コーディネーションをサポートしないサーバを使用することがあります。これには、Adaptive Server の以前のバージョンや CIS を使用して設定された Sybase 以外のデータベース・ストアも含まれます。このような状況では、strict dtm enforcement をオフに設定して、Adaptive Server がトランザクションを以前のバージョンの Adaptive Server や他のデータ・ストアに送信できるようにします。ただし、このように設定しても、これらのサーバのリモート作業がオリジナル・トランザクションとともにロールバックまたはコミットされることが保証されるわけではありません。

suspend audit when device full

要約	
デフォルト値	0 (オフ)
値の範囲	0 (オフ)、1 (オン)
ステータス	動的
表示レベル	中間
必要な役割	システム・セキュリティ担当者
設定グループ	セキュリティ関連

suspend audit when device full は、監査デバイスの空きがまったくなくなったときの Adaptive Server の動作を決定します。

注意 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。

次のいずれかの値を選択します。

- 0 − 現在の監査テーブルが満杯になったときに、次の監査テーブルをトランケートし、そのテーブルを現在の監査テーブルとして使用します。suspend audit when device full を 0 に設定すると、監査プロセスが決して中断しないことを保証できます。ただし、古い監査レコードをアーカイブしていない場合は、それらが失われる危険性があります。
- 1 − 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・プロセスが中断します。通常の操作を再開するには、システム・セキュリティ担当者がログインして、空のテーブルを現在の監査テーブルとして設定する必要があります。この間、システム・セキュリティ担当者は、通常の監査の対象外となります。通常の操作であれば監査レコードが生成されるようなアクションをシステム・セキュリティ担当者が実行すると、そのイベントに関するエラー・メッセージと情報が Adaptive Server のエラー・ログに送信されます。

syb_sendmsg port number

要約	
デフォルト値	0
有効な値	0、1024 ～ 65535、またはシステム制限値
ステータス	動的
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	ネットワーク通信

syb_sendmsg port number は、sp_sendmsg または syb_sendmsg でメッセージを UDP (ユーザ・データグラム・プロトコル) ポートに送信するときに Adaptive Server が使用するポート番号を指定します。

複数のエンジンが設定されている場合は、指定したポート番号以降の番号のポートが、エンジンごとに 1 つずつ使用されます。ポート番号をデフォルト値の 0 に設定すると、ポート番号は Adaptive Server によって割り当てられます。

注意 UDP ポートへのメッセージ送信は Windows ではサポートされていません。

UDP ポートにメッセージを送信できるようにするには、システム・セキュリティ担当者が allow sendmsg 設定パラメータを 1 に設定する必要があります。UDP のメッセージ機能を有効にするには、システム管理者が allow sendmsg を 1 に設定する必要があります。詳細については、「allow sendmsg」(83 ページ)を参照してください。UDP のメッセージ機能の詳細については、『リファレンス・マニュアル：プロシージャ』の「sp_sendmsg」を参照してください。

sysstatistics flush interval

要約	
デフォルト値	0
有効な値	0 ～ 32767
ステータス	動的
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	SQL Server 管理

sysstatiscts flush interval は、sysstatistics をフラッシュする間隔を分単位で指定します。

Adaptive Server は、DML 文の一部として変更されたテーブル内のローとカラムの数に関する統計を動的に保持し、sysstatistics flush interval の値に従ってこの統計をフラッシュします。

この統計の方が正確であるため、クエリの最適化ではこの統計が使用されます。datachange 関数は、前回の update statistics 以降に変更されたテーブル、カラム、またはパーティション・レベルのデータの量を計算し、オブジェクトの統計を更新します。

メモリ内統計は、サーバの正常なシャットダウン中に常にディスクにフラッシュされます。sysstatistics flush interval を使用して、ハウスキーパ・タスクがメモリ内統計を定期的にディスクにフラッシュする間隔を設定できます。このハウスキーピング・タスクを無効にするには、sysstatistics flush interval を 0 に設定します。

systemwide password expiration

要約	
デフォルト値	0
値の範囲	0 ～ 32767
ステータス	動的
表示レベル	中間
必要な役割	システム・セキュリティ担当者
設定グループ	セキュリティ関連

systemwide password expiration は、変更されたパスワードの有効日数を設定します。systemwide password expiration を 0 に設定すると、パスワードは無期限になります。

パスワードは指定した日数が経過すると期限切れになります。たとえば、パスワードの有効期限の間隔が 30 日である新しいログオンを 2007 年 8 月 1 日の午前 10 時半に作成したとすると、2007 年 8 月 31 日の午前 10 時半にパスワードの有効期限が切れます。

アカウントのパスワードが最後に変更されたときからの期間が *number_of_days* を超えていると、そのアカウントのパスワードは期限切れと見なされます。

期限切れまでに残っている日数が systemwide pa_ssword expiration の値の 25% または 7 日のいずれか大きい方よりも少なくなると、ユーザがログインするたびに期限切れまでに残っている日数を示すメッセージが表示されます。ユーザは期限が切れる前であればいつでも自分のパスワードを変更できます。

アカウントのパスワードの期限が切れてもユーザは Adaptive Server にログインできますが、sp_password を使用して自分のパスワードを変更するまでは、コマンドは一切実行できません。アカウントが sp_password 以外は実行できないモードのときに、システム・セキュリティ担当者がユーザのパスワードを変更すると、新しいパスワードが割り当てられた時点でそのアカウントは通常のモードに戻ります。

この制限が適用されるのは、パスワードの期限が切れた後に開始されるログイン・セッションだけです。パスワードの期限が切れた時点でログイン済みのユーザの場合は、次回ログインするときまでは影響がありません。

tape retention in days

要約	
デフォルト値	0
値の範囲	0 ～ 365
ステータス	動的
表示レベル	中間
必要な役割	システム管理者
設定グループ	バックアップとリカバリ

tape retention in days は、データベースまたはトランザクション・ログ・ダンプに使用した後、それぞれのテープを保持する日数を指定します。このパラメータを使用すると、ダンプ・テープを誤って上書きすることを避けることができます。

たとえば、tape retention in days を 7 日間に設定した場合は、そのテープに最後にダンプしてから 7 日間が過ぎる前にテープを使用しようとすると、Backup Server の警告メッセージが表示されます。

ダンプ・コマンドを実行するときに with init オプションを使用することによって、警告を無視することができます。ただし、このようにするとテープは上書きされ、テープ上のデータはすべて消失することになります。

dump database コマンドと dump transaction コマンドのどちらにも、そのダンプの tape retention in days 値を無効にする retaindays オプションがあります。『システム管理ガイド 第 2 巻』の「第 12 章 ユーザ・データベースのバックアップとリストア」を参照してください。

tcp no delay

要約	
デフォルト値	1 (オン)
有効な値	0 (オフ)、1 (オン)
ステータス	静的
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	ネットワーク通信、O/S リソース

tcp no delay は、TCP (Transmission Control Protocol) パケットのバッチ処理を制御します。デフォルト値では TCP パケットはバッチ処理されません。

TCP は、通常は物理ネットワーク・フレームをできるかぎり多くのデータで満たすために、パケットをわずかに遅延させることによって小さな論理パケットを 1 つの大きい物理パケットにまとめます。これは、ネットワークを通して送信するもののほとんどがキー入力である端末エミュレーション環境でネットワーク・スループットを改善することを目的としたものです。

しかし、小さな TDS (Tabular Data Stream) パケットを使用するアプリケーションの場合は、TCP パケットのバッチ処理を無効にすることをおすすめします。

注意 TCP パケットのバッチを無効にするということは、サイズに関係なくパケットが送られて、ネットワーク・トラフィック量が増えることを意味します。

text prefetch size

要約	
デフォルト値	16
有効な値	0 ～ 65535
ステータス	動的
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	ネットワーク通信

text prefetch size は、プリフェッチして既存のバッファ・プールに格納できる text データ、unitext データ、image データのページ数を制限します。Adaptive Server がプリフェッチするのは、Adaptive Server 12.x で作成されたか、または dbcc rebuild_text を使用してアップグレードされた text データ、unitext データ、image データだけです。

time slice

要約	
デフォルト値	100
値の範囲	50 ～ 1000
ステータス	動的
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	SQL Server 管理

time slice は、Adaptive Server のスケジューラが 1 つのタスクの実行を許可する時間をミリ秒単位で設定します。time slice の設定値が小さすぎると、Adaptive Server がタスクの切り替えに費やす時間が過大になり、応答時間が長くなります。設定値が大きすぎると、CPU 集約タスクがエンジンを独占することがあり、この場合も応答時間が長くなります。デフォルト値を使用すると、1 つのタスクが 100 ミリ秒間実行されてから、CPU が別のタスクに渡されます。

『パフォーマンス&チューニング・シリーズ：基本』の「第 3 章 エンジンと CPU の使用方法」を参照してください。

Adaptive Server エンジンによる自発的な CPU 解放への time slice の影響を調べるには、sp_sysmon を使用してください。『パフォーマンス&チューニング・シリーズ：sp_sysmon による Adaptive Server の監視』を参照してください。

total data cache size

要約	
デフォルト値	0
値の範囲	0 ～ 2147483647
ステータス	計算された値
表示レベル	基本
必要な役割	システム管理者
設定グループ	キャッシュ・マネージャ、メモリ使用

total data cache size は、データ、インデックス、ログ・ページ用に現在使用できるメモリの量をキロバイト単位で表します。このパラメータは、計算によって設定される値であり、ユーザは直接設定できません。

データ・キャッシュ用に使用できるメモリの量は、次のような多くの要因によって変わります。

- マシンで使用できる物理メモリの量
- 次のパラメータに設定される値
  - total logical memory
  - number of user connections
  - total procedure cache percent
  - number of open databases
  - number of open objects
  - number of open indexes
  - number of devices

この他の多数のパラメータも、その程度は大きくはありませんが、使用できるメモリの量に影響します。

Adaptive Server でのメモリ割り付け方法とデータ・キャッシュについては、「設定パラメータ」(76 ページ) を参照してください。

total logical memory

要約	
デフォルト値	該当なし
値の範囲	該当なし
ステータス	読み込み専用
表示レベル	中間
必要な役割	システム管理者
設定グループ	メモリ使用、物理メモリ

total logical memory は、Adaptive Server の現在の設定における論理メモリの総量を表示します。論理メモリの総量とは、Adaptive Server の現在の設定で使用されるメモリの量です。total logical memory が表すのは、確保する必要はありますが、常に使用されるとはかぎらないメモリです。特定の時点で使用されているメモリの量については、「total physical memory」を参照してください。total logical memory を使用してメモリ設定パラメータを設定することはできません。

total physical memory

要約	
デフォルト値	該当なし
値の範囲	該当なし
ステータス	読み込み専用
表示レベル	中間
必要な役割	システム管理者
設定グループ	メモリ使用

total physical memory は読み込み専用の設定パラメータで、Adaptive Server の現在の設定における物理メモリの総量を表示します。物理メモリの総量とは、特定の時点で Adaptive Server が使用しているメモリの量をいいます。max memory の値が total logical memory より大きく、total logical memory の値が total physical memory より大きくなるように、Adaptive Server を設定してください。

transfer utility memory size

要約	
デフォルト値	4096
値の範囲	0 ～ 2147483647
ステータス	動的
表示レベル	中間
必要な役割	システム管理者
設定グループ	SQL Server 管理

Adaptive Server は、transfer table コマンド用と増分転送のマークが付けられたテーブル用のメモリ・プールを維持します。このプールは、現在と過去の転送についてのステータス情報を維持するためのメモリ、および転送ファイルとの間で読み取り、書き込みを行うためのメモリを提供します。transfer utility memory size はこのメモリ・プールのサイズを指定します。

このプールの単位はメモリ・ページであり、2048 バイトのブロックです。デフォルト・サイズは、100 個を超える増分転送のマークが付けられたテーブルを収容するのに十分な大きさであり、すべてを同時に転送できます。

使用しているインストール環境に増分転送のマークが付けられたテーブルがなく、かつ transfer table コマンドが使用されない場合には、このメモリ・プールのサイズをゼロに設定してこのメモリを取り戻すこともできます。

txn to pss ratio

要約	
デフォルト値	16
有効な値	1 ～ 2147483647
ステータス	静的
表示レベル	基本
必要な役割	システム管理者
設定グループ	DTM 管理、メモリ使用

Adaptive Server はトランザクションを設定可能なサーバ・リソースとして管理します。新しいトランザクションが開始するたびに、Adaptive Server は、サーバの起動時に作成されるグローバル・プールから未使用の「トランザクション記述子」を取得する必要があります。トランザクション記述子は、Adaptive Server がアクティブなトランザクションを表すために使用する内部メモリ構造です。

Adaptive Server では、以下のものを表すために、未使用のトランザクション記述子が 1 つ必要です。

- 個々のサーバ・トランザクションの外部ブロック。トランザクションの外部ブロックは、クライアントが新規に begin transaction コマンドを実行すると明示的に作成されます。また、クライアントがトランザクションを定義する begin transaction を使用せずに Transact-SQL を使用してデータを変更する場合にも、Adaptive Server によって暗黙的に作成されます。

  注意　さらに begin transaction コマンドを使用して、ネストされたトランザクション・ブロックを作成しても、そのトランザクション・ブロック用のトランザクション記述子を取得する必要はありません。

- 「マルチデータベース・トランザクション」でアクセスされる各データベース。Adaptive Server は、新しいデータベースのデータがトランザクションによって使用または変更されるたびに、新しいトランザクション記述子を取得する必要があります。

txn to pss ratio によって、サーバで使用できるトランザクション記述子の合計数が決まります。ブート時に、この値 (比率) に PSS 構造数を乗算して、トランザクション記述子のプールが作成されます。

```
# of transaction descriptors = PSS structures * txn to pss ratio
```

デフォルト値は 16 であり、これにより 12.x より前のバージョンとの互換性が確保されます。これらのバージョンでも各ユーザ接続に 16 個のトランザクション記述子を割り付けていました。バージョン 12.x 以降では、同時に発生可能なトランザクション数は、サーバ内で使用できるトランザクション記述子の数によってのみ制限されます。

注意 1 つのユーザ・トランザクションでアクセスできるデータベース数の上限は、Adaptive Server インストール環境に存在するデータベース数と同じです。たとえば、Adaptive Server のデータベース数が 25 個ならば、ユーザ・トランザクションで 25 個のデータベースを使用することができます。

使用システムに合わせた txn to pss ratio の最適化

使用のピーク時に、sp_monitorconfig を使用してトランザクション記述子の使用状況を調査します。

```
sp_monitorconfig "txn to pss ratio"

Usage Information at date and time: Apr 22 2002  2:49PM.
Name               num_free  num_active   pct_act    Max_Used   Reused
--------------     --------  ----------  ---------   --------   ------
txn to pss ratio   784       80          10.20       523        NA
```

num_free の値がゼロまたは非常に小さい場合は、サーバ内のトランザクション記述子が解放されるまで Adaptive Server が待機するので、トランザクションが遅延する場合があります。その場合には、txn to pss ratio の値を大きくすることを検討してください。

Max_Used の値が小さすぎるときは、他のサーバ機能が使用できるはずのメモリを、未使用のトランザクション記述子が消費している可能性があります。その場合には、txn to pss ratio の値を小さくすることを検討してください。

unified login required

要約	
デフォルト値	0 (オフ)
値の範囲	0 (オフ)、1 (オン)
ステータス	動的
表示レベル	中間
必要な役割	システム・セキュリティ担当者
設定グループ	セキュリティ関連

unified login required は、Adaptive Server にログインするすべてのユーザがセキュリティ・メカニズムによって認証されることを要求します。統一化ログイン・セキュリティ・サービスを使用するには、use security services パラメータを 1 にしてください。

upgrade version

要約	
デフォルト値	15000
値の範囲	0 ～ 2147483647
ステータス	動的
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	SQL Server 管理

upgrade version パラメータは、マスタ・デバイスをアップグレードしたアップグレード・ユーティリティのバージョンをレポートします。アップグレード・ユーティリティは、アップグレード中にこのパラメータをチェックし、変更します。

警告！ upgrade version を再設定しないでください。設定を変更すると、Adaptive Server の動作に重大な問題が発生することがあります。

次のように値を指定せずに upgrade version を使用すると、マスタ・デバイスのアップグレードが行われているかどうかを調べることができます。

```
sp_configure "upgrade version"
```

use security services

要約	
デフォルト値	0 (オフ)
値の範囲	0 (オフ)、1 (オン)
ステータス	静的
表示レベル	中間
必要な役割	システム・セキュリティ担当者
設定グループ	セキュリティ関連

use security services は、Adaptive Server がネットワークベース・セキュリティ・サービスを使用することを指定します。このパラメータが 0 に設定されると、ネットワークベース・セキュリティ・サービスは使用できません。

user log cache size

要約	
デフォルト値	論理ページ・サイズ
値の範囲	2048[a] ～ 2147483647 a. 最小値はサーバの論理ページ・サイズによって決まる
ステータス	静的
表示レベル	中間
必要な役割	システム管理者
設定グループ	メモリ使用、ユーザ環境

user log cache size は、それぞれのユーザのログ・キャッシュのサイズをバイト単位で指定します。このサイズは、サーバの論理ページ・サイズによって決まります。設定されているユーザ接続とワーカー・プロセスのそれぞれに 1 つのユーザ・ログ・キャッシュが割り当てられます。Adaptive Server は、このキャッシュをユーザ・トランザクション・ログ・レコードのバッファとして使用し、これによってトランザクション・ログの終わりでの競合を減少させます。

ユーザ・ログ・キャッシュの空きがなくなるか、トランザクションの完了などの別のイベントが発生すると、Adaptive Server はユーザ・ログ・キャッシュからデータベース・トランザクション・ログにすべてのログ・レコードを「フラッシュ」します。個々のログ・レコードをすぐにデータベースのトランザクション・ログに追加するのではなく、ユーザごとのログ・キャッシュ内にいったんまとめることによって、特に複数のエンジンが設定されている SMP システムの場合に、ログへ書き込むプロセスの競合を減少させます。

注意 データとログのセグメントが分かれていないデータベースを使用するトランザクションでは、ユーザ・ログ・キャッシュは各ログ・レコードの後でトランザクション・ログにフラッシュされます。バッファリングは行われません。データベースに専用のログ・セグメントがない場合は、user log cache size の値は増やさないでください。

user log cache size の設定値は、アプリケーションの 1 つのトランザクションで書き込まれるログ情報の最大量を超えないようにしてください。Adaptive Server はトランザクションの完了時にユーザ・ログ・キャッシュをフラッシュするので、ユーザ・ログ・キャッシュに余分に割り付けられたメモリは無駄になります。4000 バイトより大きなトランザクション・ログ・レコードを生成するトランザクションがサーバ内にないのであれば、この値を超えないように user log cache size を設定します。次に例を示します。

```
sp_configure "user log cache size", 4000
```

user log cache size を大きすぎる値に設定するとメモリを浪費します。user log cache size の設定値が小さすぎると、ユーザ・ログ・キャッシュが満杯になってフラッシュすることが 1 つのトランザクションについて何度も発生し、トランザクション・ログの競合が増加します。トランザクションの量が小さい場合は、トランザクション・ログの競合は大きくはありません。

このパラメータのキャッシュ動作への影響を確認するには、sp_sysmon を使用してください。『パフォーマンス&チューニング・シリーズ：sp_sysmon による Adaptive Server の監視』を参照してください。

user log cache spinlock ratio

要約	
デフォルト値	20
値の範囲	1 ～ 2147483647
ステータス	動的
表示レベル	中間
必要な役割	システム管理者
設定グループ	メモリ使用、ユーザ環境

user log cache spinlock ratio は、複数のエンジンを実行している Adaptive Server における、ユーザ・ログ・キャッシュの「スピンロック」当たりのユーザ・ログ・キャッシュの比率を指定します。設定されているユーザ接続ごとに 1 つのユーザ・ログ・キャッシュが存在します。

デフォルト値では、サーバに設定された 20 のユーザ接続ごとに 1 つのスピンロックがあることを意味します。

このパラメータのキャッシュ動作への影響を確認するには、sp_sysmon を使用してください。『パフォーマンス&チューニング・シリーズ：sp_sysmon による Adaptive Server の監視』を参照してください。

wait event timing

要約	
デフォルト値	0
値の範囲	0 ～ 1
ステータス	動的
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	メモリ使用、モニタリング

wait event timing は、Adaptive Server で個々の待機イベントの統計を収集するかどうかを制御します。タスクは、さまざまな理由で待機させられることがあります (たとえば、バッファの読み込みの完了を待つなど)。monSysWaits テーブルには、待機イベントごとの統計が含まれます。monWaitEventInfo には、すべての待機イベントのリストが含まれます。

workload manager cache size

要約	
デフォルト値	80
有効な値	80 ～ 2147483647
ステータス	動的
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	共有ディスク・クラスタ

workload manager cache size は、Workload Manager が使用できるメモリの最大量を 2K ページ単位で指定します。『Cluster ユーザーズ・ガイド』の「第 6 章 負荷の管理」を参照してください。

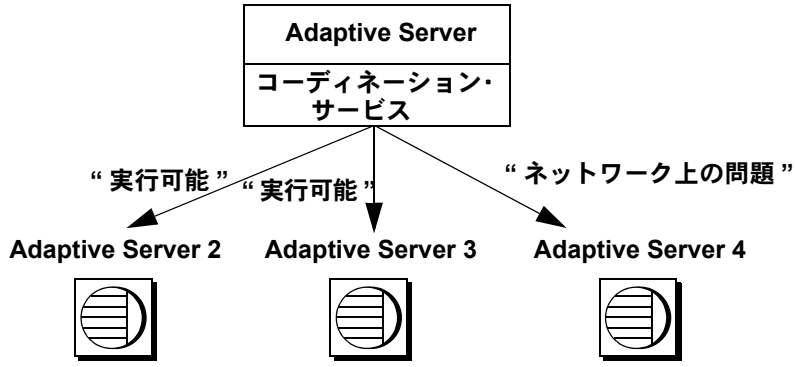
xact coordination interval

要約	
デフォルト値	60 (秒)
有効な値	1 ～ 2147483647 (秒)
ステータス	動的
表示レベル	10
必要な役割	システム管理者
設定グループ	DTM 管理

xact coordination interval は、リモート・サーバに送信されたトランザクション分岐の解決を試行する時間の間隔を定義します。

コーディネーティング・サーバである Adaptive Server は、分散トランザクションに参加しているリモート・サーバの処理の解決を定期的に試みます。図 5-6 に示すように、コーディネーティング・サーバは、分散トランザクションに参加している個々のリモート・サーバと順に交信します。さまざまな理由で、コーディネーション・サービスがトランザクション分岐を解決できない場合があります。たとえば、ネットワークに問題があるためにリモート・サーバと交信できない場合には、xact coordination level で指定される時間が経過した後で、コーディネーティング・サーバは接続を再試行します。

図 *5-6:* リモート・トランザクション分岐の解決



xact coordination interval をデフォルト値の 60 に設定すると、Adaptive Server は 1 分おきにリモート・トランザクションの解決を試みます。この値をこれより小さくすると、分散トランザクションの処理速度が向上する可能性があります。ただし、そのトランザクション自体が 1 分以内に解決できる場合にかぎります。通常の状況では、xact coordination interval の値を小さくすることによってパフォーマンスが損なわれることはありません。

xact coordination interval の値を大きくすると、分散トランザクションの処理速度が低下し、トランザクション分岐によるリソースの保持時間が通常よりも長くなる場合があります。通常は、xact coordination interval の値をデフォルト値よりも大きくしないでください。

xp_cmdshell context

要約	
デフォルト値	1
有効な値	0, 1, 2
ステータス	動的
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	拡張ストアド・プロシージャ

xp_cmdshell context は、xp_cmdshell システム ESP を使用して実行されるオペレーティング・システム・コマンドのセキュリティ・コンテキストを設定します。コンテキストの値は、コマンドを実行するアカウントを指定します。

- 0 − コマンドが XP Server のアカウントで実行される
- 1 − コマンドがユーザのアカウントで実行される
- 2 − ユーザがシステム管理者権限を持っている場合にかぎり、コマンドが XP Server のアカウントで実行される

xp_cmdshell context を 1 に設定すると、xp_cmdshell セキュリティ・コンテキストは、オペレーティング・システム・レベルのアカウントを持っているユーザに制限されます。動作はプラットフォームによって異なります。xp_cmdshell context が 1 に設定されている場合に xp_cmdshell ESP を使用するには、Adaptive Server のユーザ名と同じ名前のオペレーティング・システム・ユーザ・アカウントが存在している必要があります。たとえば、Adaptive Server のユーザ名が “sa” のユーザは、“sa” というオペレーティング・システム・レベルのユーザ・アカウントを持っていなければ xp_cmdshell を使用することはできません。

Windows では、xp_cmdshell context が 1 に設定されている場合に xp_cmdshell が正常に実行されるのは、Adaptive Server にログインしているユーザ名が、Adaptive Server が稼働しているシステムの Windows システム管理者権限を持つ有効な Windows ユーザ名である場合だけです。

他のプラットフォームでは、xp_cmdshell context が 1 に設定されている場合に xp_cmdshell が正常に実行されるのは、Adaptive Server を起動したユーザに、オペレーティング・システム・レベルの “superuser” 権限がある場合だけです。Adaptive Server は、xp_cmdshell の実行要求を受け取ると、この ESP を要求したユーザ名の uid を調べ、その uid のパーミッションでオペレーティング・システムのコマンドを実行します。

xp_cmdshell context が 0 の場合は、xp_cmdshell からオペレーティング・システム・コマンドを実行するときに使用されるのは Adaptive Server が稼働しているオペレーティング・システム・アカウントのパーミッションです。これによって、ユーザ自身のオペレーティング・システム・アカウントのセキュリティ・コンテキストでは通常実行できないオペレーティング・コマンドを実行できるようになります。

第 6 章

ディスク・リソースについての概要

Adaptive Server の記憶領域管理のさまざまなプロパティ (データベース、テーブル、インデックスを配置する場所とそれぞれに割り付ける領域のサイズなど) のほとんどは、妥当なデフォルト値が設定されています。記憶領域の割り付けと管理は集中管理されることも多く、Adaptive Server に対するディスク・リソースの割り付けと、割り付けたディスク・リソースへのデータベース、テーブル、インデックスの物理的な配置については、通常はシステム管理者が最終的な制御権を持っています。

デバイスの割り付けとオブジェクトの配置

新しいシステムを設定するとき、システム管理者は、必要なディスク・リソースの数とサイズに直接影響するさまざまな問題を検討する必要があります。この問題とは、Adaptive Server にディスク・リソースを追加するコマンドやプロシージャをどのように実行するかということです。

表 *6-1:* デバイスの割り付けに関するトピック

作業	章
データベース・デバイスのデフォルト・プールの初期化と割り付け	「第 7 章 データベース・デバイスの初期化」
リカバリを目的としたデータベース・デバイスのミラーリング	『システム管理ガイド 第 2 巻』の「第 2 章 データベース・デバイスのミラーリング」

Adaptive Server の初期ディスク・リソースを割り付けた後で、システム管理者、データベース所有者、オブジェクト所有者は、データベースとデータベース・オブジェクトをどのデータベース・デバイスに配置するかを検討する必要があります。オブジェクトの配置を具体的に検討することによって、データベース・オブジェクトを、使用しているシステムのどこに常駐させるか、また、オブジェクトにデバイスを共有させるかどうかを決定します。オブジェクトを配置する作業については、表 6-2 に示す各章を含むこのマニュアル全体で説明しています。

表 6-2: オブジェクトの配置に関するトピック

作業	章
データベースを特定のデータベース・デバイスに配置する	『システム管理ガイド 第 2 巻』の「第 6 章 ユーザ・データベースの作成と管理」
テーブルとインデックスを特定のデータベース・デバイスに配置する	『システム管理ガイド 第 2 巻』の「第 6 章 ユーザ・データベースの作成と管理」

デバイスの割り付けを検討するときは、オブジェクトの配置と切り離して考えないでください。たとえば、あるテーブルを 2 つで一組の専用のデバイスに配置する場合は、最初にその 2 つのデバイスを Adaptive Server に割り付けます。この章の各項ではデバイスの割り付けとオブジェクトの配置の両方にかかわる問題に共通する概要を説明し、参照情報がある場合は参照先の章を示しています。

ディスク・リソースの管理に使用するコマンド

表 6-3 は、Adaptive Server にディスク・リソースを割り付けるためにシステム管理者が使用する主なコマンドのリストと、そのコマンドについて説明している参照先の章をまとめたものです。

表 6-3: ディスク・リソースの割り付けに使用するコマンド

コマンド	作業	参照箇所
disk init name = "*dev_name*" physname = "*phys_name*"...	特定の Adaptive Server で使用できるように物理デバイスを設定する。データベース・デバイス名 (*dev_name*) を割り当てる。これは、他の Adaptive Server コマンド内でこの物理デバイスを指定するときに使用される。	「第 7 章 データベース・デバイスの初期化」
sp_deviceattr *logicalname*, *optname*, *optvalue*	既存のデータベース・デバイス・ファイルの *dsync* 設定を変更する。	「第 7 章 データベース・デバイスの初期化」
sp_diskdefault "*dev_name*"...	*dev_name* をデフォルトのデータベース領域の汎用プールに追加する。	「第 7 章 データベース・デバイスの初期化」
disk resize name = "*device_name*" size = *additional_space*	データベース・デバイスのサイズを動的に拡大する。	「第 7 章 データベース・デバイスの初期化」
disk mirror name = "*dev_name*" mirror = "*phys_name*"...	特定の物理デバイス上のデータベース・デバイスをミラーリングする。	『システム管理ガイド 第 2 巻』の「第 2 章 データベース・デバイスのミラーリング」第 2 巻

表 6-4 は、オブジェクトの配置に使用するコマンドのリストです。オブジェクトの配置がパフォーマンスに及ぼす影響については、『パフォーマンス＆チューニング・シリーズ：物理データベースのチューニング』の「第 1 章 データの物理的配置の制御」を参照してください。

表 *6-4:* ディスク・リソース上のオブジェクトの配置に使用するコマンド

コマンド	作業	参照箇所
create database...on *dev_name* または alter database...on *dev_name*	特定の Adaptive Server データベースで使用できるようにデータベース・デバイスを設定する。create database の log on 句によって、このデータベースのログを特定のデータベース・デバイスに配置することを指定する。	『システム管理ガイド第 2 巻』の「第 6 章 ユーザ・データベースの作成と管理」
create database... または alter database...	デフォルト・データベース・デバイス上の領域を割り付けるには、on *dev_name* 句を指定しないでこれらのコマンドを実行する。	『システム管理ガイド第 2 巻』の「第 6 章 ユーザ・データベースの作成と管理」
sp_addsegment *seg_name*, *dbname*, *devname* および sp_extendsegment *seg_name*, *dbname*, *devname*	特定のデータベースで使用できるデバイスから、セグメント (領域の集合に名前を付けたもの) を作成する。	『システム管理ガイド第 2 巻』の「第 8 章 セグメントの作成と使用」
create table...on *seg_name* または create index...on *seg_name*	データベース・オブジェクトを作成して、データベースに割り当てられたディスク領域の特定セグメントに配置する。	『システム管理ガイド第 2 巻』の「第 8 章 セグメントの作成と使用」
create table... または create index...	データベースに割り付けられた領域の汎用プール (デフォルト・デバイス) にテーブルやインデックスを配置するには、on *seg_name* を指定しないでこれらのコマンドを実行する。	『システム管理ガイド第 2 巻』の「第 8 章 セグメントの作成と使用」

記憶領域の管理に関する考慮事項

システム管理者は、Adaptive Server データベースへの領域の物理的な割り付けに関するさまざまな事項について決定する必要があります。この場合に考慮しなければならない重要な事項は、次のとおりです。

- リカバリ － ディスク・ミラーリングと、別の物理デバイス上でのログ保管という 2 つの機能によって、物理的なディスク障害が起きた場合でも完全なリカバリを実現できます。
- パフォーマンス － ディスクの読み込み／書き込み速度が非常に重要であるテーブルやデータベースについては、データベース・オブジェクトを物理デバイスに適切に配置することによってパフォーマンスが向上します。ディスク・ミラーリング機能を使用すると、ディスクの書き込み速度が低下します。

リカバリ

リカバリ機能は、複数のディスク・デバイスの使用を決定する主な要因となります。データベース・デバイスをミラーリングすると、ノンストップ・リカバリが可能になります。また、別の物理デバイス上にデータベースのログを保管することによって、完全なリカバリが保証されます。

別デバイスでのログの保管

データベース・デバイスをミラーリングしていない場合に完全なリカバリができるようにするには、データベースのトランザクション・ログを、データベース内の実際のデータ (インデックスも含む) とは別のデバイスに保管する必要があります。ハード・ディスクの障害が起きても、ログは別のデバイスに安全に保存されているので、データベースのダンプをロードしてからこのログ・レコードを適用することによって、最新のデータベースを作成できます。create database の log on 句の詳細については、『システム管理ガイド 第 2 巻』の「第 6 章 ユーザ・データベースの作成と管理」を参照してください。

ミラーリング

ハード・ディスクの障害が発生した場合に確実にノンストップ・リカバリできるようにするには、Adaptive Server の全デバイスを別の物理ディスクにミラーリングします。『システム管理ガイド 第 2 巻』の「第 2 章 データベース・デバイスのミラーリング」を参照してください。

パフォーマンス

ログとデータベース・オブジェクトを別々のデバイスに配置すると、システムのパフォーマンスを向上させることができます。

- テーブルをあるハード・ディスクに配置し、ノンクラスタード・インデックスを別のハード・ディスクに配置すると、処理が 2 台のディスク・ドライブに分割されるため、物理的な読み込みと書き込みが速くなります。
- サイズの大きなテーブルを 2 台のディスクに分割すると、特にマルチユーザ・アプリケーションのパフォーマンスが向上します。
- ログとデータが同じデバイスに配置されている場合は、トランザクション・ログ・レコードのユーザ・ログ・キャッシュ・バッファリングは行われません。
- テーブルを分割すると、ヒープ・テーブルに複数の挿入ポイントが作成され、並列クエリ処理を実行するように設定されているシステムの並列処理度が増加します。また、テーブルの I/O を複数のデータベース・デバイス間に分散できます。

オブジェクトの配置がパフォーマンスに及ぼす影響については、『パフォーマンス&チューニング・シリーズ：基本』の「第 1 章 データの物理的配置の制御」を参照してください。

インストール時のステータスおよびデフォルト設定

インストール・プログラムとスクリプトによって、マスタ・デバイスが初期化され、master、model、sybsystemprocs、sybsecurity の各データベースとテンポラリ・データベースが設定されます。

Adaptive Server をインストールすると、システム・データベース、システム定義のセグメント、データベース・デバイスは、次のような構成で設定されます。

- master、model、tempdb の各データベースは、マスタ・デバイス上にインストールされます。
- sybsystemprocs データベースは、指定のデバイス上にインストールされます。
- 各データベースに system、default、logsegment の 3 つのセグメントが作成されます。
- マスタ・デバイスが、ユーザが作成するすべてのデータベース用のデフォルト記憶デバイスとなります。

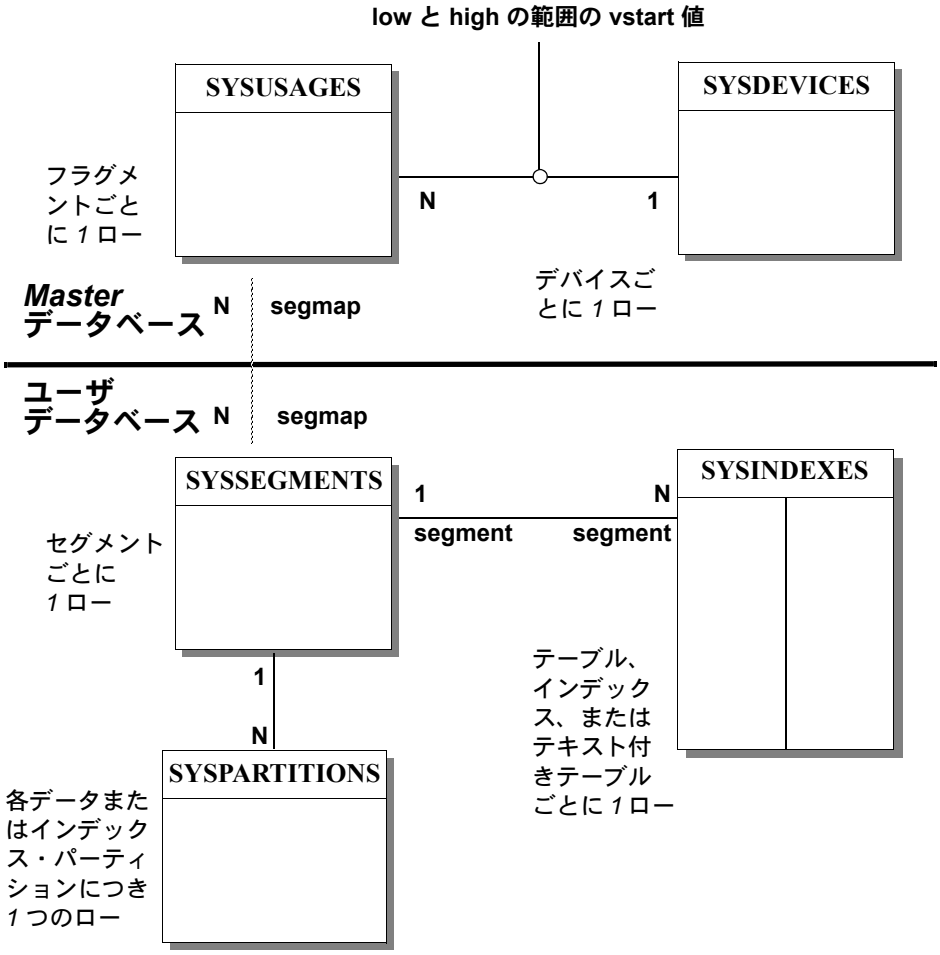
  注意 デフォルトの記憶領域用の新しいデバイスを初期化した後は、sp_diskdefault を使用してマスタ・デバイスをデフォルトの記憶領域から除外してください。ユーザのデータベースやオブジェクトをマスタ・デバイスに格納しないでください。詳細については、「デフォルト・デバイスの指定」(275 ページ) を参照してください。

- 監査データベース　sybsecurity　をインストールする場合は、このデータベースは専用のデバイス上に配置されます。

記憶領域を管理するシステム・テーブル

master データベース内の 2 つのシステム・テーブル sysusages および sysdevices と、各ユーザ・データベース内の 3 つのシステム・テーブル syssegments、sysindexes、および syspartitions は、データベース、テーブル (トランザクション・ログ・テーブル syslogs を含む)、およびインデックスの配置を記録します。これらのテーブルの関係を図 6-1 に示します。

図 *6-1:* 記憶領域を管理するシステム・テーブル



sysdevices テーブル

master データベース内の sysdevices テーブルには、「データベース・デバイス」ごとにローが 1 つ格納されます。また、Adaptive Server で使用できるダンプ・デバイス (テープ、ディスク、またはオペレーティング・システム・ファイル) ごとに 1 つのローが格納されることがあります。

disk init コマンドを実行すると、データベース・デバイスに対応するエントリが master..sysdevices に追加されます。sp_addumpdevice を使用して追加されるダンプ・デバイスについては、『システム管理者ガイド 第 2 巻』の「第 11 章 バックアップおよびリカバリ・プランの作成」を参照してください。

sysdevices には、デバイスごとに次の 2 つの名前が格納されます。

- 論理名またはデバイス名：以降のすべての記憶領域管理コマンドで使用される名前です。この名前は、sysdevices の name カラムに保管されます。通常は、ユーザにわかりやすい名前を付け、できればデバイスの用途を示すような名前にします (たとえば、“logdev” や “userdbdev”)。
- 物理デバイス名：オペレーティング・システムでのこのデバイスの実際の名前です。この名前を使用するのは disk init コマンドだけです。それ以降の Adaptive Server のデータ記憶領域コマンドはすべて論理名を使用します。

データベースまたはトランザクション・ログを 1 つまたは複数のデバイス上に配置するには、create database 文または alter database 文内でデバイスの論理名を指定します。完全なリカバリを確実に実行できるようにするには、create database に log on 句を指定して、データベースのトランザクション・ログを別のデバイス上に配置します。また、log on を使用するには、sysdevices 内でログ・デバイスのエントリが割り当てられている必要があります。

1 つのデータベースを 1 つまたは複数のデバイス上に配置できます。また、1 つのデバイスに 1 つ以上のデータベースを保管することもできます。『システム管理ガイド 第 2 巻』の「第 6 章 ユーザ・データベースの作成と管理」を参照してください。

sysusages テーブル

master データベース内の sysusages テーブルには、Adaptive Server のデータベースに割り当てられているすべての領域が記録されています。

create database と alter database を実行すると、データベース・デバイスまたはデバイス・フラグメントごとに 1 つのローが sysusages に追加され、これによって新しい領域がデータベースに割り当てられます。create または alter database によってデバイス上の領域の一部だけが割り付けられた場合に、その部分をフラグメントと呼びます。

sp_addsegment、sp_dropsegment、sp_extendsegment を実行すると、デバイスに対応する sysusages テーブル内の segmap カラムが変更され、デバイスがセグメントにマッピングされるか、マッピングが解除されます。『システム管理ガイド 第 2 巻』の「第 8 章 セグメントの作成と使用」を参照してください。

syssegments テーブル

各データベースに 1 つずつ存在する syssegments テーブルには、データベース内のセグメントが登録されています。「セグメント」とは、特定のデータベースで使用できるデータベース・デバイスやフラグメントの集合のことです。テーブルやインデックスは、特定のセグメントに割り当てる、つまり特定の物理デバイスに割り当てることも、複数の物理デバイスの集合に割り当てることもできます。

create database コマンドは、syssegments テーブル内にデフォルトのエントリを作成します。sp_addsegment と sp_dropsegment は、syssegments テーブルのエントリの追加と削除を行います。

sysindexes テーブル

sysindexes テーブルには、各テーブルとインデックスが登録されています。また、各テーブル、クラスタード・インデックス、ノンクラスタード・インデックスが存在するセグメントや、テキスト・ページのチェーンが存在するセグメントも記録されています。さらに、そのテーブルまたはインデックスに対する max_rows_per_page 設定など、その他の情報も記録されています。

create table、create index、alter table の各コマンドは、sysindexes 内に新しいローを作成します。テーブルを分割すると、sysindexes 内のそのテーブルに対応するエントリの機能が変化します。

syspartitions テーブル

syspartitions テーブルには、各テーブルとインデックス・パーティションが登録されています。また、そのパーティションが存在するセグメントも記録されています。syspartitions は、記憶領域の管理に関する重要な情報 (データ・ページ・チェーンまたはインデックス・ページ・チェーンの最初のページ、ヒープの最後のページ、インデックス・パーティションのルート・ページなど) を保持します。

create table、create index、alter table の各コマンドは、syspartitions 内に新しいローを作成します。

第 7 章

データベース・デバイスの初期化

データベース・デバイス

データベース・デバイスには、データベースを構成するオブジェクトが格納されます。「デバイス」という語は、1 つの物理デバイスを指すとは限りません。データベースやそのオブジェクトを保管するための、ディスクの一部 (ディスク・パーティションなど) やファイル・システム内のファイルをデバイスと呼ぶこともあります。

データベース・デバイスまたはファイルをデータベース記憶領域として使用するには、Adaptive Server によって認識されるように準備する必要があります。この処理を「初期化」と呼びます。

初期化されたデータベース・デバイスに対しては、次の処理が実行できます。

- create database コマンドと alter database コマンドで使用できるようにデフォルト・デバイス・プールに割り当てる。
- ユーザ・データベースが使用できる領域のプールに割り当てる。
- ユーザ・データベースに割り当てて、1 つまたは複数のデータベース・オブジェクトの保管に使用する。
- データベースのトランザクション・ログを保管するために割り当てる。

disk init コマンドの使用

システム管理者は、disk init コマンドを使用して、新しいデータベース・デバイスを初期化します。このコマンドは次の処理を行います。

- 指定された物理ディスク・デバイスまたはオペレーティング・システム・ファイルをデータベース・デバイス名にマッピングする。
- master..sysdevices に新しいデバイスを登録する。
- そのデバイスをデータベース記憶領域として使用できるように準備する。

注意 disk init を実行する前に、プラットフォームの『ASE インストール・ガイド』を参照して、データベース・デバイスの選択方法と、Adaptive Server で使用するためにそのデータベース・デバイスを準備する方法を確認してください。Sybase データベースのパフォーマンスを最高にするために、コンピュータのディスク・パーティションの再設定が必要な場合もあります。

---

disk init コマンドは、データベース・デバイスを複数の「アロケーション・ユニット」に分割します。アロケーション・ユニットは、256 論理ページから構成される 1 つのグループです。アロケーション・ユニットのサイズは、サーバに設定されている論理ページ・サイズ (2、4、8、16K) によって異なります。disk init コマンドは、各アロケーション・ユニットの最初のページをアロケーション・ページとして初期化します。そのアロケーション・ユニットにデータベースがある場合は、その情報がこのページに保管されます。

警告！ disk init コマンドの実行後は、必ず master データベースのダンプを行ってください。このようにすれば、master が損傷した場合でも、簡単かつ確実にリカバリできます。『システム管理ガイド 第 2 巻』の「第 13 章 システム・データベースのリストア」を参照してください。

---

disk init の構文

disk init の構文については、『リファレンス・マニュアル：コマンド』を参照してください。

論理デバイス名の指定

device_name は、有効な識別子でなければなりません。この名前は、create database コマンドと alter database コマンド、およびセグメントを管理するシステム・プロシージャで使用します。論理デバイス名は、Adaptive Server だけが認識するもので、サーバが実行されるオペレーティング・システムには認識されません。

物理デバイス名の指定

データベース・デバイスの *physname* には、ロー・ディスク・パーティション (UNIX) の名前、外部デバイスの名前、またはオペレーティング・システム・ファイルの名前を指定します。プラットフォームが PC の場合は、通常、*physname* にはオペレーティング・システム・ファイルの名前を指定します。

デバイス番号の選択

Adaptive Server は disk init vdevno パラメータを受け付けますが、このパラメータは必須ではありません。vdevno を指定する場合、1 ～ 2,147,483,647 の識別子から現在未使用のものを選択できます (仮想デバイス ID 0 は master デバイスで使用しています)。たとえば、`vdevno = 33` を指定すると、デバイスに仮想デバイス ID 33 が割り当てられます。vdevno を指定しない場合、sysdevices に登録されている最大の vdevno より大きい数字が選択されます。

作成可能なデータベース・デバイス数は、number of devices 設定パラメータによって制限されます。Adaptive Server の初期設定では、デバイス数は 10 です。sp_configure を使用して、number of devices の値を変更します。

インストールされているシステム上で同時に使用可能なデバイス数が、オペレーティング・システムによって制限されている場合もあります。オペレーティング・システムでは、各 Sybase デバイスが 1 つのオープン・ファイルと見なされます。

Adaptive Server は、次に使用できるデータベース・デバイス識別番号を自動的に指定します。これが、vdevno (仮想デバイス番号、virtual device number) です。disk init コマンドを発行するときに、この番号を指定する必要はありません。

手動で vdevno を選択する場合は、Adaptive Server によって使用されるデバイス間でユニークな番号を指定してください。デバイス番号 0 は、マスタ・デバイスを表します。有効な番号は 1 ～ 2,147,483,647 です。この値の範囲内で未使用の vdevno を選択可能です。

既に vdevno に使用されている番号を確認するには、sp_helpdevice からのレポートの vdevno カラムを調べるか、または次のクエリを実行し、現在使用中のデバイス番号をすべてリストします。

```
select vdevno from master..sysdevices
   where status & 2=2
```

ここで、`status & 2=2` は物理ディスクを示します。

デバイス・サイズの指定

デバイスのサイズを指定するには、キロバイトを示す場合は ‘k’ または ‘K’、メガバイトを示す場合は ‘m’ または ‘M’、ギガバイトを示す場合は ‘g’ または ‘G’、テラバイトを示す場合は ‘t’ または ‘T’ を使用します。disk init コマンドと create database コマンドのどちらも、必ず単位指定子を入力することをおすすめします。これは、実際に割り付けられるページ数との混同を避けるためです。単位指定子は、一重引用符、二重引用符、または角カッコで囲んでください。

理論上は、最大 2,147,483,647 個のディスク・デバイスを作成し、各デバイスに最大 2,147,483,648 の 2K ブロックを割り当てることができます。最大インストール可能サイズが、データベース・サイズ、ハードウェア、およびオペレーティング・システムの事実上の制限になります。

次のガイドラインが、disk init の構文に適用されます。

- disk init または disk reinit の size 引数で単位を指定しなかった場合、size はデフォルトで仮想ページの数と解釈されます。たとえば、`size = 15000` と入力すると、Adaptive Server では 15,000 仮想ページと解釈されます。仮想ページは 1 ページあたり 2048 バイトです。
- disk resize コマンドを使用して既存のデータベース・デバイスのサイズを増やすことはできますが、減らすことはできません。
- 新しいデータベースを作成するための新しいデバイスの最小サイズ (size) は、サーバが使用する論理ページ・サイズによって異なります。詳細については、表 7-1 を参照してください。

表 *7-1:* 最小データベース・サイズ

論理ページ・サイズ	最小データベース・サイズ
2K	3MB
4K	6MB
8K	12MB
16K	24MB

データベースを model データベースより小さくすることはできません。model データベースが上記の最小サイズより大きい場合、そのサイズが最小データベース・サイズになります。

Adaptive Server はアロケーション・ユニット (256 論理ページから成るグループ) のデータベース領域を割り付けおよび管理します。create database を使用して作成可能な最小データベースは 1 MB であるため、使用可能なデータベース・デバイスの最小サイズは 1 MB と 256 論理ページのどちらか大きい方になります (論理ページ・サイズが 2K または 4K の場合は 1 MB、論理ページ・サイズが 8K の場合は 2 MB、論理ページ・サイズが 16K の場合は 4 MB)。

デバイスのサイズを決めるときは、この 256 ページ単位のグループを考慮すると、領域の無駄を避けることができます。たとえば、インストールされているシステムで 16k の論理ページ・サイズを使用している場合、デバイス・サイズを `size = '31M'` に指定すると、アロケーション・ユニットが 4MB になるため、デバイスの最後の 3MB が無駄になります。

ロー・デバイスを初期化する場合は、プラットフォームの『ASE インストール・ガイド』を参照して、オペレーティング・システムで使用可能なデバイス・サイズを調べてください。使用可能な合計サイズは、使用するプラットフォームの最大値までです。いったん Adaptive Server 用にディスクを初期化すると、そのロー・デバイスの領域は別の目的では使用できなくなります。

disk init は、size を使用して sysdevices.high の最終仮想ページ番号の値を計算します。sysdevices.high および sysdevices.low の値は、2K バイトのブロック (Adaptive Server の物理ディスク管理の単位) で構成される仮想ページのページ番号です。この値は、インストール環境の論理ページ・サイズとは異なることがあります。

注意　size パラメータで指定したブロック数が物理デバイスにない場合、disk init コマンドは実行できません。オプションの vstart パラメータを使用する場合は、vstart パラメータと size パラメータで指定したブロック数の合計が物理デバイス上に必要です。このブロック数を確保できない場合は、このコマンドは異常終了します。

---

dsync 設定の指定 (オプション)

UNIX のオペレーティング・システム・ファイル上で初期化されたデバイスの場合に、dsync 設定は、そのファイルへの書き込みをバッファリングするかどうかを制御します。dsync 設定がオンの場合、Adaptive Server は UNIX の dsync フラグを使用してデータベース・デバイス・ファイルを開きます。このフラグを使用すると、デバイス・ファイルへの書き込みが物理記憶メディアに対して直接行われるようになり、システム障害が発生した場合もデバイス上の Adaptive Server のデータをリカバリすることができます。

dsync がオフの場合は、UNIX のファイル・システムによってデバイス・ファイルへの書き込みがバッファリングされることもありますが、デバイス上のデータのリカバリは保証されません。dsync は、データ整合性が要求されていない場合にのみオフにしてください。

注意　ロー・パーティション上で初期化されたデバイスの場合、dsync 設定は無視されます。代わりに、データベース・デバイスへの書き込みは、物理メディアに対して直接行われます。

パフォーマンスへの *dsync* の影響

データベース・デバイス・ファイルで dsync 設定を使用すると、いくつかのパフォーマンスのトレードオフが発生します。

- Adaptive Server は、HP-UX のオペレーティング・システム・ファイルに対する非同期 I/O をサポートしません。
- データベース・デバイス・ファイルに dsync オプションが設定されている場合は、デバイス・ファイルに書き込む Adaptive Server エンジンは、書き込みオペレーションが完了するまで待機します。このため、更新操作中は、パフォーマンスが低下する場合があります。
- dsync がオンの場合、データベース・デバイス・ファイルへの書き込みオペレーションは、以前のバージョンの Adaptive Server (dsync がサポートされていない場合) より遅くなることがあります。これは、Adaptive Server が、キャッシュ・データを UNIX ファイル・システムのバッファにコピーする代わりに、データをディスクに書き込まなくてはならないからです。

  最高の書き込みパフォーマンスが要求される場合 (ただし、システム障害後のデータ整合を必要としない場合)、dsync をオフにすると、以前のバージョンの Adaptive Server と同様のパフォーマンスを得ることができます。たとえば、専用のデバイス・ファイル上に tempdb を保管する場合に、dsync を使用しているときのパフォーマンスが許容範囲内にない場合は、dsync を無効に設定します。
- 読み込み操作の応答時間は、一般的に、デバイスを UNIX オペレーティング・システム・ファイル上に保管する方が、ロー・パーティション上に保管するよりも良くなります。デバイス・ファイルからのデータは、Adaptive Server のキャッシュだけでなく、UNIX ファイル・システムのキャッシュも利用できるので、物理ディスクにアクセスせずにより多くの読み込みを実行できる可能性があります。

dsync の制限事項

dsync を使用するときは、次の制限事項が適用されます。

- マスタ・デバイスの場合、dsync は、常に true に設定され、変更することはできません。
- sp_deviceattr プロシージャを使用してデバイス・ファイルの dsync 設定を変更するときは、変更内容を有効にするために Adaptive Server を再起動します。
- バージョン 12.x より前の Adaptive Server からアップグレードしたときに dsync が true に設定されるのは、マスタ・デバイス・ファイルのみです。他のデバイス・ファイルの dsync 設定を変更するには、sp_deviceattr を使用してください。

- ロー・パーティションに保管されるデータベース・デバイスに対する dsync 設定は無視されます。ロー・パーティションに保管されるデバイスへの書き込みは、常に物理メディアに対して直接行われます。
- directio パラメータと dsync パラメータは互いに排他的です。デバイスの dsync を true に設定した場合、同じデバイスの directio を true に設定することはできません。デバイスの directio を有効にするには、dsync の設定を false に変更しておく必要があります。

directio によるオペレーティング・システム・バッファの回避

disk init、disk reinit、sp_deviceattr の directio パラメータを指定することにより、オペレーティング・システムのバッファ・キャッシュを回避して、データをディスクに直接転送することができます。directio は、I/O の方法やパフォーマンス向上の効果の点ではロー・デバイスと同じですが、ロー・デバイスより使いやすく、ファイル・システム・デバイスの管理が容易です。directio は、マスタ・デバイスに設定することはできません。directio は、静的パラメータであるため、このパラメータを有効にするには、Adaptive Server を再起動します。

注意　directio は、すべてのプラットフォームで使用できるわけではありません。サポートされていないプラットフォームで、directio パラメータを使用して disk init を発行すると、Adaptive Server は、`[No such parameter: 'directio']` というメッセージを発行します。

---

directio オプションのデフォルト値は、すべてのプラットフォームで false (オフ) に設定されます。

注意　リカバリが重要ではないデータベース (tempdb など) に使用するデバイスの dsync は、デフォルトで false に設定されている場合があります。このようなデバイスの directio を有効にすると、パフォーマンスが低下することがあるため、デバイスの用途を十分に検討してから directio を有効にしてください。

次の例では、directio を使ってデータをディスクに直接書き込む “user_disk” という名前のデバイスが作成されます。

```
disk init
name = "user_disk",
physname = "/usr/u/sybase/data/userfile1.dat",
size = 5120, directio= true
```

UNIX オペレーティング・システムのファイルでディスクの 10MB を初期化するには、次のように入力します。

```
disk reinit
name = "user_disk",
physname = "/usr/u/sybase/data/userfile1.dat",
size = 5120, directio= true
```

デフォルトでは、既存のすべてのデバイスの directio が無効に設定されているので、sp_deviceattr を使用して有効にします。

```
sp_deviceattr device_name, directio, [true | false]
```

たとえば、次のコマンドを実行すると、“user_disk” デバイスの directio によるディスク書き込みが有効になります。

```
sp_deviceattr user_disk, directio, true
```

disk init のその他のオプション・パラメータ

vstart は、Adaptive Server がデータベース・デバイスの使用を始める開始仮想アドレス、つまりオフセットです。vstart に指定できる単位指定子は、k または K (キロバイト)、m または M (メガバイト)、g または G (ギガバイト)、t または T (テラバイト) です。オフセットのサイズは、vstart の値を入力する方法によって異なります。

- 単位を指定しなかった場合、vstart は 2K ページをその開始アドレスとして使用します。たとえば、`vstart = 13` と指定すると、Adaptive Server では 13 × 2K ページが開始アドレスのオフセットとして使用されます。
- 単位指定子によってサイズを指定すると、vstart は、入力した値を開始アドレスとして使用します。たとえば、`vstart = "13M"` と指定すると、Adaptive Server では 13MB で開始アドレスのオフセットが設定されます。

vstart のデフォルト値 (および通常は優先値) は 0 です。指定したデバイスで、vstart + size の合計ブロックが利用できない場合は、disk init コマンドは失敗します。

オプションの cntrltype キーワードは、ディスク・コントローラを指定します。デフォルト値は 0 です。システム管理者からの指示があった場合にだけ、このオプションを再設定してください。

注意 ディスクの初期化を実行するには、Adaptive Server を起動したユーザが、初期化するデバイスに対する適切なオペレーティング・システム・パーミッションを所有している必要があります。

デバイス情報の表示

sp_helpdevice を実行すると、sysdevices テーブル内のデバイスに関する情報が表示されます。

デバイス名なしで sp_helpdevice を実行すると、Adaptive Server で使用可能なすべてのデバイスがリストされます。デバイス名を指定すると、そのデバイスに関する情報がリストされます。次の例では、sp_helpdevice を使用してマスタ・デバイスに関する情報を表示します。

```
                    sp_helpdevice master
device_name  physical_name  description
-----------  -------------- ------------------------------------------
master        d_master         special, default disk, physical disk, 30 MB


status      cntrltype    vdevno     vpn_low   vpn_high
------      ----------   -------    -------   ---------
3           0            0          0         10239
```

master..sysdevices 内の各ローが表す情報は、次のとおりです。

- データベースのバックアップに使用されるダンプ・デバイス (テープ、ディスク、またはファイル)。
- データベース記憶領域として使用されるデータベース・デバイス。

sysdevices の初期内容は、オペレーティング・システムによって異なります。通常、sysdevices には次のエントリが含まれます。

- マスタ・デバイスのエントリ (1 つ)。
- sybsystemprocs データベースのエントリ (1 つ)。これは pubs2 や sybsyntax のような追加のデータベースの保管用や、ユーザ・データベースとログ用に使用できます。
- テープのダンプ・デバイスのエントリ (2 つ)。

監査プログラムをインストールした場合には、sybsecurity 用の別のデバイスもあります。

vpn_low と vpn_high の各カラムは、デバイスに割り当てられているページ番号を表します。ダンプ・デバイスの場合、これらのカラムはデバイスのメディア容量を表します。

status フィールドは、デバイスのタイプ、ユーザがデータベース・デバイスを指定しないで create database コマンドまたは alter database コマンドを発行したときにディスク・デバイスがデフォルトの記憶デバイスとして使用されるかどうか、ディスク・ミラーリング情報、および dsync 設定値を表します。

表 7-2: *sysdevices* 内のステータス・ビット

ビット	意味
1	デフォルトのディスク (create database コマンドまたは alter database コマンドでロケーションが指定されない場合に使用できる)
2	物理ディスク
4	論理ディスク (使用しない)
8	スキップ・ヘッダ (テープ・ダンプ・デバイスで使用)
16	ダンプ・デバイス
32	逐次書き込み
64	デバイスがミラーリングされている
128	読み込みがミラーリングされている
256	セカンダリ・ミラーリング側のみ
512	ミラーリング使用可能
2048	内部で使用 (disk unmirror, side = retain の後に設定)
4096	ミラーリングを解除する必要があるプライマリ・デバイス (内部で使用)
8192	ミラーリングを解除する必要があるセカンダリ・デバイス (内部で使用)
16384	UNIX ファイル・デバイスは、dsync 設定を使用する (物理メディアに直接書き込みが行われる)

ダンプ・デバイスと sp_addumpdevice の詳細については、『システム管理者ガイド 第 2 巻』の「第 11 章 バックアップおよびリカバリ・プランの作成」を参照してください。

デバイスの削除

データベース・デバイスとダンプ・デバイスを削除するには、sp_dropdevice を使用します。

```
sp_dropdevice logicalname
```

データベースが使用しているデバイスは削除できません。まず、データベースを削除してください。

sp_dropdevice は、sysdevices からデバイス名を削除します。sp_dropdevice を実行しても、オペレーション・システム・ファイルは削除されるのではなく、Adaptive Server からアクセスできなくなるだけです。sp_dropdevice を使用した後でファイルを消去するには、オペレーティング・システムのコマンドを使用してください。

デフォルト・デバイスの指定

Adaptive Server のユーザによるデータベースの作成時に使用されるデフォルト・データベース・デバイスのプールを作成するには、デバイスを初期化した後で sp_diskdefault システム・プロシージャを実行します。sp_diskdefault を実行すると、sysdevices 内で、そのデバイスにデフォルト・デバイスのマークが付けられます。ユーザがデータベース・デバイスを指定せずにデータベースを作成 (または変更) すると、デフォルト・ディスク領域プールから新しいディスク領域が割り付けられます。

sp_diskdefault の構文は次のとおりです。

```
sp_diskdefault logicalname, {defaulton | defaultoff}
```

ユーザ・デバイスを追加した後、デフォルト領域のプールからマスタ・デバイスを削除するには、defaultoff オプションを使用します。

```
sp_diskdefault master, defaultoff
```

次のコマンドは、sprocdev (sybsystemprocs データベースが格納されているデバイス) を、デフォルト・デバイスとして指定します。

```
sp_diskdefault sprocdev, defaulton
```

デフォルト・デバイスは複数指定することもできます。その場合、デバイスは sysdevices テーブルに表示される順番 (つまりアルファベット順) に使用されます。最初のデフォルト・デバイスがいっぱいになると、2 番目以降のデフォルト・デバイスが順に使用されます。

注意　初期化したデータベース・デバイスを、デフォルト・デバイス・プールではなく、特定のデータベースまたはデータベース・オブジェクトに割り当てることもできます。たとえば、テーブルのサイズが特定のデバイスのサイズを超えないようにする場合です。

デフォルト・デバイスと非デフォルト・デバイスの選択

sp_diskdefault を利用すると、ユーザによるデータベースの作成や変更が可能な状態を維持しながら、パフォーマンスとリカバリを考慮した上で領域の使用について計画を立てることができます。

次のデバイスはデフォルト・デバイスとして使用しないでください。

- マスタ・デバイス
- sybsecurity 用のデバイス
- ログ専用のデバイス
- 高パフォーマンスのデータベースが常駐するデバイス

sybsystemprocs を保持しているデバイスは、他のユーザ・データベース用に使用できます。

注意 ディスク・ミラーリングまたはセグメントを使用している場合、どのデバイスをデフォルトのリストに追加するかを決めるときには、注意が必要です。通常、ミラーリングするデバイスや、セグメント上にオブジェクトが配置されるデータベースには、デフォルト領域の一部ではなく、特定のデバイスを割り付けてください。

disk resize コマンドによるデバイスのサイズ拡大

disk resize コマンドを使用すれば、新しいデバイスを初期化することなく、データベース・デバイスのサイズを動的に増やすことができます。たとえば、/sybase/testdev.dat に 10MB の追加領域が必要になった場合には、disk resize コマンドを実行して、必要な領域をデバイスに割り付けることができます。この追加領域は、create database コマンドと alter database コマンドで使用できます。

disk resize コマンドを使用してサイズを拡大できるのは、ロー・パーティション上のデバイスとファイル・システムのデバイスです。デバイス領域の拡大量として指定できる最小値は、1MB または 1 アロケーション・ユニットのいずれか大きい方です。

ページ・サイズ	アロケーション・ユニットのサイズ	最小追加サイズ
2K	0.5MB	1MB
4K	1MB	1MB
8K	2MB	2MB
16K	4MB	4MB

ダンプとロードに使用するデバイスには、disk resize コマンドは使用できません。

デバイスに設定されたプロパティは、サイズを拡大した後も設定は解除されずに残ります。つまり、デバイスに dsync が設定されている場合、このデバイスのサイズを拡大した後も dsync は設定されたままになります。また、デバイスのサイズを拡大する前に設定されていたアクセス権限もそのまま残ります。

sa_role を持つユーザが disk resize コマンドを実行すると、次の処理が行われます。

- master....sysdevices の high の値を更新する
- データベース記憶領域の追加領域を作成する

disk resize コマンドの監査証跡を使用すると、デバイスのサイズが変更された回数を追跡できます。サイズ変更対象のデバイスは常にオンラインになり、サイズ変更の処理中でもユーザはデバイスを使用できます。

ディスクのサイズ変更には、次のような要件があります。

- デバイスが disk init コマンドで初期化されている。
- *device_name* は、有効な論理デバイス名である。
- サイズ変更操作の実行中はミラーリングを無効にする。サイズ変更操作の完了後に、ミラーリングを再開できます。

この例で、デバイス testdev の設定は次のとおりです。

```
sp_helpdevice testdev
device_name  physical_name          description
  status  cntrltype  vdevno          vpn_low           vpn_high
-----------  -------------------    -------------
  ------- ---------  -------------   -------------     --------------
testdev      /sybase/dev/testdev.dat   special, dsync on, directio off,
physical disk, 10.00MB
  16386    0           1                    0              5119
```

disk resize コマンドを使用してデバイス testdev のサイズを 4MB 増やすには、次のように入力します。

```
disk resize
name = "test_dev",
size = "4M"
```

testdev.dat のサイズは、14MB に拡大されています。

```
sp_helpdevice testdev
device_name  physical_name          description
  status  cntrltype  vdevno          vpn_low           vpn_high
-----------  -------------------    -------------
  ------- ---------  -------------   -------------     --------------
testdev      /sybase/dev/testdev.dat   special, dsync on, directio off,
physical disk, 14.00MB
  16386           0         1          0                  7167
```

disk resize の構文については、『リファレンス・マニュアル：コマンド』を参照してください。

ディスク領域の不足

ディスクの物理的な初期化中にディスク領域不足のエラーが発生した場合は、エラーが発生する前の時点での利用可能な最大サイズにデータベース・デバイスが拡張されます。

たとえば、4K の論理ページを使用するサーバでデバイスのサイズを 40MB 増やそうとしたときに、空きディスク領域が 39.5MB しかない場合は、デバイスは 39.5MB 分だけ拡張されます。

disk resize コマンドで、デバイスのサイズの縮小はできません。

第 8 章

データベース・オプションの設定

データベース・オプションは、データベースの動作について次のような制御を行います。

- トランザクションの動作
- テーブル・カラムのデフォルトの設定
- ユーザ・アクセスの制限
- リカバリと bcp 操作の実行
- ログの動作

システム管理者とデータベース所有者は、データベース・オプションを使用して、データベース全体の設定を行うことができます。この点では、データベース・オプションは、サーバ全体に影響する sp_configure パラメータや、現在のセッションまたはストアド・プロシージャにだけ影響する set オプションとは異なります。

sp_dboption プロシージャの使用

データベース全体の設定内容を変更するには、sp_dboption を使用します。オプションは変更されるまで有効なままです。sp_dboption プロシージャの機能は、次のとおりです。

- パラメータが指定されないときは、データベース・オプションの全リストを表示する。
- パラメータが指定されたときは、データベース・オプションを変更する。

ユーザ・データベースに対してのみ、オプションを変更できます。master データベースのオプションは変更できません。ユーザ・データベース内のデータベース・オプションを変更 (または、データベース・オプションのリストを表示) するには、master データベースを使用している状態で sp_dboption を実行します。

構文は次のとおりです。

sp_dboption [*dbname*, *optname*, {true | false}]

注意 model のデータベース・オプションを変更しても、Adaptive Server を再起動したときに、tempdb、または現在のユーザが定義した複数のテンポラリ・データベースには影響を与えません。これらの変更は、model データベースを変更した後に作成されたデータベースにのみ表示されます。Adaptive Server を再起動すると、テンポラリ・データベースに含まれていたオブジェクトとデータはクリアされますが、データベース・オプションはリセットされません。

データベース・オプションの説明

master データベースに対するアクセス権を持つすべてのユーザは、パラメータを付けずに sp_dboption を実行すれば、データベース・オプションのリストを表示できます。sp_dboption からのレポートは、次のように表示されます。

```
sp_dboption
Settable database options.
--------------------
abort tran on log full
allow nulls by default
async log service
auto identity
dbo use only
ddl in tran
delayed commit
identity in nonunique index
no chkpt on recovery
no free space acctg
read only
select into/bulkcopy/pllsort
single user
trunc log on chkpt
trunc. log on chkpt.
unique auto_identity index
```

特定のデータベースに設定されているオプションに関するレポートを表示する場合は、そのデータベース内で sp_helpdb システム・プロシージャを実行します。

「setuser コマンド」参照各データベース・オプションの詳細については、『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。

データベースの各オプションの表示

特定のデータベースに設定されているオプションを確認するには、sp_helpdb を使用します。sp_helpdb の出力では、アクティブなオプションが “status” カラム内に表示されます。

次の出力例は、mydb データベースでは read only オプションが on になっていることを示します。

```
sp_helpdb mydb

name           db_size  owner  dbid   created          status
-----           -------   -----  ----   ------------     --------------------
mydb           20.0 MB    sa     5    Mar 05, 2005     read only

device_fragments      size    usage             created        free kbytes
-----------------     ------  ------------      --------       -----------
master                10.0 MB  data and log      Mar 05 2005           1792

device                          segment
------------------------------ ------------------------------
master                          default
master                          logsegment
master                          system
```

すべてのデータベースのオプションをまとめて表示するには、データベースを指定しないで sp_helpdb を発行します。

```
sp_helpdb
name            db_size  owner dbid   created            status
-------------   -------- ----- ----   -----------        -------------------
master          48.0 MB   sa      1   Apr 12, 2005       mixed log and data
model            8.0 MB   sa      3   Apr 12, 2005       mixed log and data
pubs2           20.0 MB   sa      6   Apr 12, 2005       select into/
     bulkcopy/pllsort, trunc log on chkpt, mixed log and data
sybsystemdb      8.0 MB   sa      5   Apr 12, 2005       mixed log and data
sybsystemprocs 112.0 MB   sa      4   Apr 12, 2005       trunc log on chkpt,
     mixed log and data
tempdb           8.0 MB   sa      2   Apr 12, 2005       select into/
     bulkcopy/pllsort, trunc log on chkpt, mixed log and data
```

第 9 章

文字セット、ソート順、言語の設定

この章では、Adaptive Server の国際化とローカライゼーション・サポートについて説明します。

国際化とローカライゼーションの概要

「国際化」とは、アプリケーションを複数の言語や文化的慣例に対応させることをいいます。

国際化されたアプリケーションでは、実行時に外部ファイルを使用して、言語固有の情報を表示します。このようなアプリケーションは、言語固有のコードを含んでいないので、コードに変更を加えることなくどのようなネイティブ言語の環境にも配備できます。ソフトウェア製品の 1 つのバージョンを複数の言語や地域に適応させることができ、設計変更を行わなくても各地域の要件や習慣に適合させることが可能です。このようなソフトウェア開発のアプローチは、アプリケーションのライフタイム全体を通して大幅な時間と費用の節約につながります。

「ローカライゼーション」とは、国際化された製品を特定の言語 (たとえばスペイン語) または地域の要件に合うように適応させることで、これには、その国の言語に翻訳されたシステム・メッセージ、ユーザ・インタフェースの翻訳版、その国で使用されている正しいフォーマットでの日付、時間、通貨の表示が含まれます。ソフトウェア製品の 1 つのバージョンについて、いくつものローカライズ版を作成することができます。

Sybase は、国際化とローカライゼーションの両方をサポートしています。Adaptive Server には、西欧、東欧、中東、ラテン・アメリカ、アジアにおける主な商用語の、データ処理サポートに必要な文字セット定義ファイルとソート順定義ファイルがあります。

Sybase 言語モジュールは、中国語 (簡体字)、フランス語、ドイツ語、日本語、韓国語、ブラジルで使用するポルトガル語、スペイン語の各言語に翻訳されたシステム・メッセージとフォーマットを提供します。デフォルトでは、Adaptive Server には英語のメッセージ・ファイルが付属しています。

この章では、文字セットと言語モジュールについて簡単に説明するとともに、Adaptive Server のデフォルトの文字セット、ソート順、またはメッセージ言語を変更するために必要な手順を説明します。

国際化されたシステムの利点

アプリケーションを他国でも使用できるように設計することは、きわめて面倒な仕事に思えます。プログラマたちは、国際化とは各国の文化や言語上の慣習によって必要となる部分を個々にハードコードすることだと考えがちです。

しかし、もっと良いアプローチがあります。それは、国際化されたアプリケーション、つまり実行時にローカルのコンピューティング環境を調べて使用する言語を決定し、その言語に関する情報が記述されているファイルを読み込むアプリケーションを作成することです。

国際化されたアプリケーションであれば、同じアプリケーションをどの国でも使用できます。このアプローチには次のような利点があります。

- アプリケーションを 1 つだけ作成すればよく、各国語版を個々に作成する必要はありません。
- アプリケーションの提供先の国が増えたときも、アプリケーションに変更を加える必要はありません。その国のローカライゼーション・ファイルを添付するだけで済みます。
- すべてのサイトで機能と動作を統一できます。

サンプル国際化システム

国際化されたシステムでは、国際化されたクライアント・アプリケーション、ゲートウェイ、サーバをさまざまなネイティブ言語環境の複数のプラットフォームで実行できます。

たとえば、次のようなコンポーネントで構成される国際化システムもあります。

- ニューヨーク、メキシコシティ、パリにある発注処理アプリケーション (Client-Library アプリケーション)
- ドイツにある在庫管理サーバ (Adaptive Server)
- フランスにある発注遂行アプリケーション (Adaptive Server)
- 日本にある中央会計アプリケーション (Adaptive Server と相互稼働する Open Server アプリケーション)

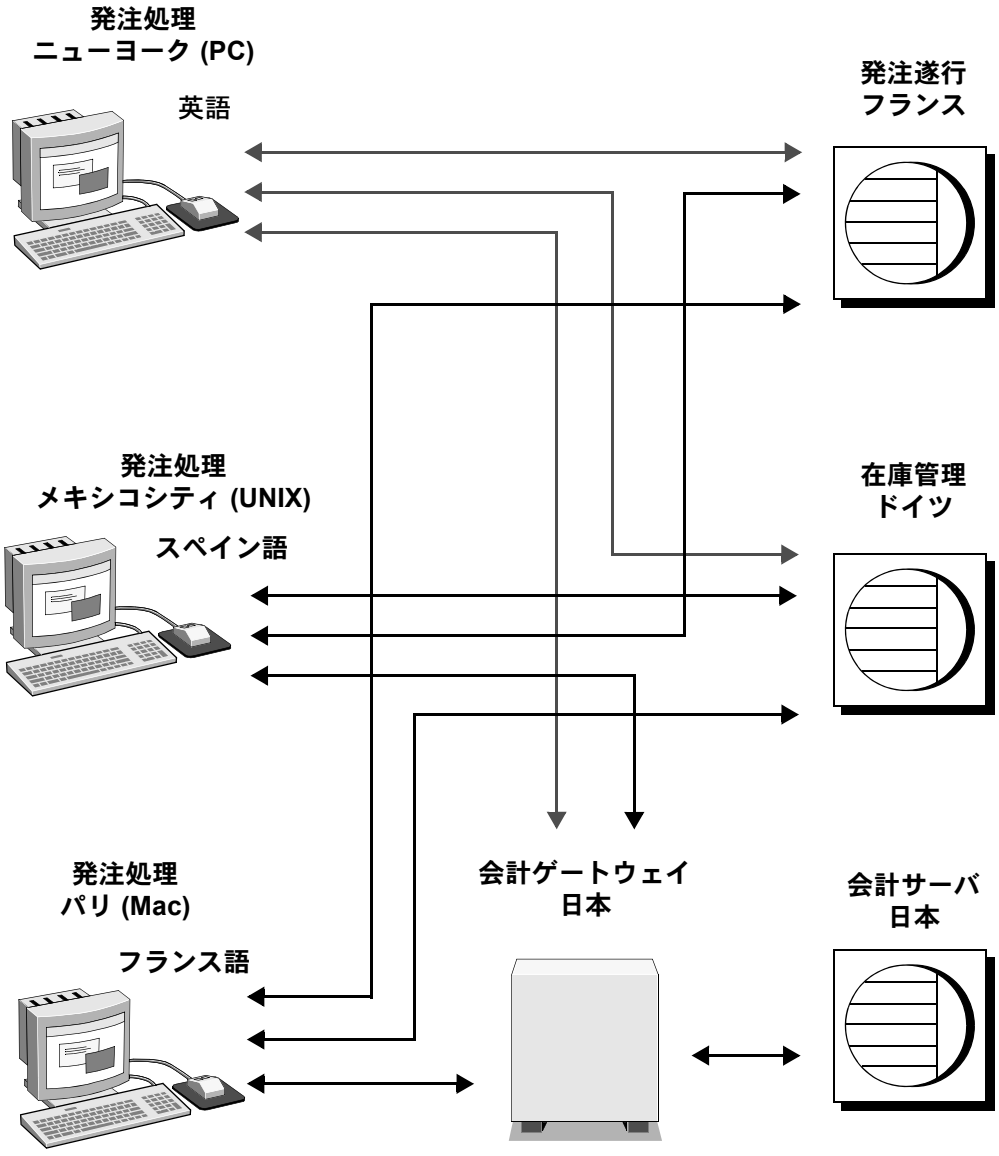
このシステムの発注処理アプリケーションは、以下の処理を実行します。

- 在庫管理サーバに対してクエリを発行し、注文された商品の在庫があるかどうかを調べる。
- 発注遂行サーバに発注データを送る。
- 財務情報を会計アプリケーションに送る。

在庫管理サーバと発注遂行サーバはクエリに応答し、会計アプリケーションは財務データを収集してレポートを作成します。

このシステムを図で表すと次のようになります。

図 *9-1:* 国際化システムの例



この例では、すべてのアプリケーションとサーバがそれぞれの国の言語と文字セットを使用して、入力データを受け取り、メッセージを出力します。

国際化システムの要素

国際化環境のサーバ言語を設定するときには、以下に挙げる 3 つの要素について操作が可能です。これらの要素について検討したうえで、クライアント／サーバ・ネットワークの構築計画を入念に作成してください。

- 文字セット － サーバがクライアント・サーバとの間でデータを送受信するときに使用する言語。すべてのクライアント・サーバについて言語上の必要事項を慎重に計画、分析したうえで、文字セットを選択してください。
- ソート順 － ソート順オプションは選択した言語および文字セットによって異なります。
- システム・メッセージ － Sybase が提供する言語のいずれか 1 つでメッセージが表示されます。提供された言語の中にサーバ言語がない場合は、システム・メッセージはデフォルト言語の英語で表示されます。

以下の項では、これらの各要素について詳しく説明します。

サーバの文字セットの選択

サーバ内では、データは特別なコードでコード化されます。たとえば、“a” という文字は 10 進数の “97” にコード化されます。「文字セット」とは、特定の文字集合 (アルファベット、数字、記号、非表示の制御文字を含む) と、それらが割り当てられた数値、つまりコードのことです。文字セットには、一般にその言語の全字母用の文字が含まれます。たとえば、英語で使用されるラテン・アルファベット、またはロシア語、セルビア語、ブルガリア語などの言語で使用されるキリル文字のような書体です。言語のサブセット (たとえば西欧言語) をサポートする、プラットフォーム固有の文字セットを「ネイティブ文字セット」といいます。Adaptive Server に付属する文字セットは、Unicode UTF-8 を除いてすべてネイティブの文字セットです。

「スクリプト」とは、書記体系のことで、たとえばラテン語、日本語、アラビア語などの言語を書き表すときにその言語を特徴付けるすべての要素の集合です。1 つのアルファベットまたはスクリプトでサポートされる言語に応じて、1 つの文字セットは 1 つ以上の言語をサポートします。たとえば、ラテン・アルファベットは西欧言語をサポートします (表 9-1 (288 ページ) のグループ 1 を参照)。それに対し、日本語というスクリプトがサポートするのは 1 つの言語 (日本語) だけです。このように、グループ 1 の文字セットはさまざまな言語をサポートしますが、多くの文字セット (グループ 101 など) は 1 つの言語しかサポートしません。

1 つの文字セットによって処理される 1 つ以上の言語を、「言語グループ」と呼びます。言語グループには、複数の言語が属することも 1 つの言語しか属さないこともあります。ネイティブ文字セットとは、特定の言語グループに属する言語の文字の、プラットフォーム固有のコード化です。

クライアント／サーバ・ネットワークでは、複数の言語でのデータ処理をサポートできます。ただし、それらの言語はすべて同じ言語グループに属していなければなりません (表 9-1 (288 ページ) を参照)。たとえば、サーバ内のデータをグループ 1 の文字セットでコード化する場合は、フランス語、ドイツ語、イタリア語など、グループ 1 に属する言語のデータであれば同じデータベースに格納できます。ただし、異なる言語グループに属する言語のデータを同じデータベースに格納することはできません。たとえば、日本語のデータはフランス語やドイツ語のデータとは一緒に保存できません。

前述のネイティブ文字セットと異なり、「**Unicode**」は国際文字セットで、日本語、中国語、ロシア語、フランス語、ドイツ語など世界中の 650 以上もの言語をサポートします。Unicode を使用すると、プラットフォームに関係なく、同じサーバ内で多数の言語グループのさまざまな言語を一緒に使用できます。詳細については、「Unicode」(290 ページ) を参照してください。

すべての文字セットがラテン・スクリプトと英語をサポートするため、それぞれの文字セットは少なくとも 2 つの言語 (英語ともう 1 つ別の言語) をサポートします。

多くの言語は複数の文字セットによってサポートされます。ある言語に対してどの文字セットをインストールするかは、クライアントのプラットフォームとオペレーティング・システムに応じて決定します。

Adaptive Server は次の言語と文字セットをサポートします。

表 *9-1:* サポートされる言語と文字セット

言語グループ	言語	文字セット
グループ 1	**西欧：** アルバニア語、カタロニア語、デンマーク語、オランダ語、英語、フェロー語、フィンランド語、フランス語、ガリシア語、ドイツ語、アイスランド語、アイルランド語、イタリア語、ノルウェー語、ポルトガル語、スペイン語、スウェーデン語	ASCII 8、CP 437、CP 850、CP 860、CP 863、CP 1252[a]、ISO 8859-1、ISO 8859-15、Macintosh Roman、ROMAN8、ROMAN9、ISO-15、CP 858
グループ 2	**東欧：** クロアチア語、チェコ語、エストニア語、ハンガリー語、ラトヴィア語、リトアニア語、ポーランド語、ルーマニア語、スロヴァキア語、スロヴェニア語 (および英語)	CP 852、CP 1250、ISO 8859-2、Macintosh Central European
グループ 4	バルト語 (および英語)	CP 1257
グループ 5	**キリル：** ブルガリア語、ベラルーシ語、マケドニア語、ロシア語、セルビア語、ウクライナ語 (および英語)	CP 855、CP 866、CP 1251、ISO 8859-5、Koi8、Macintosh Cyrillic
グループ 6	アラビア語 (および英語)	CP 864、CP 1256、ISO 8859-6

言語グループ	言語	文字セット
グループ 7	ギリシャ語 (および英語)	CP 869、CP 1253、GREEK8、ISO 8859-7、Macintosh Greek
グループ 8	ヘブライ語 (および英語)	CP 1255、ISO 8859-8
グループ 9	トルコ語 (および英語)	CP 857、CP 1254、ISO 8859-9、Macintosh Turkish、TURKISH8
グループ 101	日本語 (および英語)	CP 932 DEC Kanji、EUC-JIS、Shift-JIS
グループ 102	簡体字中国語 (PRC) (および英語)	CP 936、EUC-GB、GB18030
グループ 103	繁体字中国語 (ROC) (および英語)	Big 5、CP 950[b]、EUC-CNS、Big 5 HKSCS
グループ 104	韓国語 (および英語)	EUC-KSC、CP949
グループ 105	タイ語 (および英語)	CP 874、TIS 620
グループ 106	ベトナム語 (および英語)	CP 1258
Unicode	650 以上の言語	UTF-8

a. CP 1252 は、0x80 ～ 0x9F のコード・ポイントが CP 1252 内の文字にマップされることを除き、ISO 8859-1 と同一。
b. CP 950 は Big 5 と同一。

注意 すべての文字セットが英語をサポートするのは、どの文字セットにも最初の 128 (10 進数) 文字にラテン・アルファベット (“ASCll-7” として定義される) が含まれているからです。先頭の 128 文字より後の文字は文字セットによって異なり、別のネイティブ言語の文字をサポートするために使用されます。たとえば、CP 932 と CP 874 のどちらも、コード・ポイント 0 ～ 127 は英語とラテン・アルファベットをサポートしますが、CP 932 のコード・ポイント 128 ～ 255 は日本語文字をサポートし、CP 874 のコード・ポイント 128 ～ 255 はタイ語の文字をサポートします。

欧州通貨記号「ユーロ」をサポートする文字セットは、CP 1252 (西欧)、CP 1250 (東欧)、CP 1251 (キリル)、CP 1256 (アラビア語)、CP 1253 (ギリシャ語)、CP 1255 (ヘブライ語)、CP 1254 (トルコ語)、CP 874 (タイ語)、ISO 15、Roman 9、および CP 858 です。Unicode UTF-8 は、次の言語もサポートします。

- 繁体字中国語 (Windows および Solaris プラットフォーム)
- アラビア語、ヘブライ語、タイ語、ロシア語 (Linux プラットフォーム)

注意 iso_1 と ISO 8859-1 は名前が異なるだけで、同じ文字セットです。

異なる言語グループの言語を混在させるためには、Unicode を使用する必要があります。サーバの文字セットが Unicode であれば、650 を超える数の言語を 1 つのサーバでサポートでき、あらゆる言語グループの言語を混在させることができます。

Unicode

Unicode は、世界のすべての言語を同じデータ・セット内でコード化することを可能にした初めての文字セットです。Unicode が導入される前は、たとえば中国語でデータを保存するには、この言語に適した文字セットを選択しなければならず、他のほとんどの言語は排除されていました。同じデータ・セット内で複数の文字セットを混在させること、つまり、さまざまな言語を混在させることは不可能でした。

Unicode は 3 つのデータ型の形式でサポートされます。3 つのデータ型とは、unichar、univarchar、および unitext です。これらのデータ型は、Unicode UTF-16 でデータをコード化して保存します。

UTF-16 は、Unicode スカラ値を単一の 16 ビット値で表すコード化です (まれに、16 ビット値のペアで表すこともあります)。この 3 つのコード化は、あらゆる Unicode 文字を表すことができるという点では同じです。サーバのデフォルト文字セットとしての UTF-16 ではなく、データ型としての UTF-16 が選択されたことにより、既存のデータベース・アプリケーションを容易に、段階的に移行できます。

Adaptive Server は、SQL クエリでの Unicode リテラルと、UTF-8 のさまざまなソート順をサポートしています。

Adaptive Server が使用する文字セット・モデルは、サーバ全体で使用される、単一の設定可能な文字セットに基づいています。「文字」データ型 (char、varchar、nchar、nvarchar、text) のいずれかを使用して Adaptive Server に保存されるデータはすべて、この文字セットで記述されているものと見なされます。ソート順は、各国語に変換されたサーバ・メッセージの集合である言語モジュールと同様に、この文字セットを使用して定義されます。

クライアント・アプリケーションは、接続ダイアログの中で自身のネイティブ文字セットと言語を宣言します。設定が正しく行われていれば、サーバは、サーバの文字セットとクライアントの文字セットの間でそれ以降のすべての文字データの変換を試みます (文字データには、データベースに保存されるすべてのデータのほかに、クライアントのネイティブ言語でのサーバ・メッセージも含まれます)。このことが機能するには、サーバの文字セットとクライアントの文字セットが互いに変換可能であることが条件です。文字が他方の文字セットで定義されていない場合は、正しく機能しません。たとえば、日本語に使用される文字セットである SJIS と、ロシア語などのキリル文字言語に使用される KOI8 の場合です。このように変換不可能なことがある場合に、Unicode を使用します。Unicode は、他のすべての文字セットの文字の定義が含まれる、文字のスーパーセットと考えることができます。

Unicode データ型である unichar、univarchar、および unitext は、従来の文字セット・モデルからは完全に独立しています。クライアントは、その他の文字データの送受信とは無関係に、Unicode データを送受信します。

文字セットのインストール

Adaptive Server バージョン 12.5.1 以降では、4 バイト形式の UTF-8 がサポートされています。この形式は、16 ビット値のペア (「サロゲート・ペア」) として UTF-16 で表される、使用頻度の低い Unicode 文字を表すのに使用されます。バージョン 12.5.1 より前の Adaptive Server では、3 バイト形式の UTF-8 だけがサポートされていました。バージョン 12.5.1 より前の Adaptive Server に UTF-8 文字セットがインストールされている場合は、4 バイト形式の UTF-8 を使用できるように文字セットを再インストールする必要があります。

設定パラメータ

Unicode の UTF-16 コード化には、使用頻度の低い文字を表すための 16 ビット値のペア (「サロゲート・ペア」) が含まれています。Adaptive Server には、サロゲート・ペアの整合性を保証するための機能が組み込まれています。このチェック機能をオフにするには設定パラメータ “enable surrogate processing” を 0 に設定します。このようにすると、パフォーマンスはやや向上しますが、サロゲート・ペアの整合性は保証されなくなります。

Unicode では、「正規化」も定義されています。正規化とは、単一の文字について可能なすべての表現を、単一の表現に変換するプロセスです。基本文字とそれに続く分音記号の組み合わせには、多くの場合、等価な事前結合済み文字がありますが、両者のビット・パターンは異なっています。たとえば、次の 2 つのシーケンスは等価です。

```
0x00E9  -- é (LATIN SMALL LETTER E WITH ACUTE)

0x00650301  -- e (LATIN SMALL LETTER E), ´ (COMBINING ACUTE ACCENT)
```

enable unicode normalization 設定パラメータは、Adaptive Server が受け取った Unicode データを正規化するかどうかを制御します。

default Unicode sortorder を “binary” に設定し、enable Unicode normalization 設定パラメータを 1 に設定すると、パフォーマンスが飛躍的に向上します。この組み合わせのとき、Adaptive Server は Unicode データの性質についていくつかの仮定を行いますが、その仮定を利用するようにコードが実装されています。

関数

char 型のパラメータを受け取る関数は、すべて unichar 型も受け入れるようにオーバロードされています。複数のパラメータを持つ関数が呼び出されたときに、指定されたパラメータのいずれかが unichar 型であるときは、unichar 型以外のパラメータはすべて unichar 型に暗黙的に変換されます。

enable surrogate processing が 1 (デフォルト) に設定されているとき、サロゲート・ペアの整合性を保証するために、文字列関数ではサロゲート・ペアを分割するような処理は回避されます。位置は、サロゲート・ペアの開始位置を指すように変更されます。

unichar 型のサポートを補完するために関数が追加されています。追加された関数の中に、to_unichar() と uscalar() があります。これらの機能は、char() と ascii() に似ています。関数 uhighsurr() と ulowsurr() を利用すると、ユーザ・コードでサロゲート・ペアを明示的に扱うことができます。

関数で unitext を使用する場合には制限があります。詳細については、各関数の「使用法」のセクションにある制限の説明を参照してください。

unichar カラムの使用

isql ユーティリティまたは bcp ユーティリティを使用するとき、Unicode 値は 16 進形式で表示されます。ただし、Jutf8 フラグを使用してクライアントの文字セットが UTF-8 であることを指定した場合を除きます。この場合、サーバから受信した Unicode データはすべて UTF-8 に変換されます。次に例を示します。

```
% isql -Usa -P -Jiso_1
1> select unicode_name from people where unicode_name = 'Jones'
2> go

unicode_name
-----------------------------------------------------------------|
0x004a006f006e00650073
(1 row affected)
```

これが、次のようになります。

```
% isql -Usa -P -Jutf8
1> select unicode_name from people where unicode_name = 'Jones'
2> go

unicode_name
------------------------------------------------------------
Jones
(1 row affected)
```

これによって、アドホック・クエリが実行しやすくなります。端末ウィンドウによってはすべての Unicode 文字を表示できないものもありますが、ASCII 文字を扱う単純なテストは非常に簡単になります。

unitext の使用

可変長の unitext データ型は、Unicode 文字で最大 1,073,741,823 文字 (2,147,483,646 バイト) まで保持できます。unitext は、text データ型を使用できる場所であれば、同じセマンティックで使用できます。unitext カラムは、Adaptive Server のデフォルト文字セットとは関係なく、UTF-16 エンコーディングで保管されます。

Open Client の相互運用性

Open Client ライブラリでデータ型 cs_unichar がサポートされるようになりました。このデータ型は、短い整数の配列として宣言されたユーザ変数にバインドできます。Open Client のこのデータ型は、サーバの unichar 型、unitext 型、および univarchar 型への直接のインタフェースとなります。

Java の相互運用性

内部 JDBC ドライバにより、SQL コンテキストと Java コンテキストの間で unichar データが効率的に転送されます。

SQL から Java への場合は、クラス java.sql.ResultSet の多数の “get” メソッドを使用して、結果セットのカラムからデータを取り出します。この get メソッドは、unichar、unitext、または univarchar として定義されたカラムに対して機能します。メソッド getString() は、変換を実行する必要がないので、特に効率的といえます。

Java から SQL への場合は、クラス java.sql.PreparedStatement の setString() メソッドを使用します。内部 JDBC ドライバは、unichar、unitext、または univarchar として定義された SQL パラメータに Java の文字列データを直接コピーします。

外部 JDBC ドライバ (jConnect) は、内部ドライバと同様のシームレスなインタフェースをサポートするように変更されました。

制限事項

Adaptive Server の以前のリリースには Unicode ベースの言語パーサがなく、新しい Unicode データ型の使用には制限がありました。新しいデータ型を使用するにはサーバのデフォルト文字セットを UTF-8 として設定する必要がありましたが、この制限は、Adaptive Server バージョン 12.5.1 以降では取り除かれ、サーバのデフォルト文字セットと関係なく Unicode のデータ型を使用できるようになりました。

サーバのデフォルト文字セットの選択

サーバを設定するときに、デフォルト文字セットを指定する必要があります。デフォルトの文字セットとは、サーバがデータを格納および操作するときに使用する文字セットのことです。デフォルト文字セットは各サーバに 1 つだけ設定できます。

デフォルトでは、インストール・ツールはオペレーティング・システムのネイティブ文字セットをサーバのデフォルト文字セットと見なします。ただし、Adaptive Server がサポートする文字セットであれば、サーバのデフォルト文字セットとして選択できます (表 9-1 (288 ページ) を参照)。

たとえば、AIX が稼働する IBM RS/6000 にサーバをインストールする場合に、西欧言語の 1 つを選択すると、ISO 8859-1 がデフォルト文字セットと見なされます。

Unicode サーバをインストールする場合は、デフォルト文字セットとして UTF-8 を選択します。

Unicode サーバ以外の場合は、クライアント・システムの大部分が使用するプラットフォームを特定し、そのプラットフォームの文字セットをサーバのデフォルト文字セットとして使用します。

これには次に挙げる 2 つの利点があります。

- 文字セット間でマッピング不可能な文字を最小限にできる。

  通常は、2 つの文字セット間で文字が完全に一対一で対応することはないため、データの消失が起こる可能性があります。このことは通常はごく小さな問題でしかありません。変換されない文字のほとんどは、プラットフォームに固有であるか、あまり使用されない特殊文字だからです。

- 必要な文字セット変換を最小限にできる。

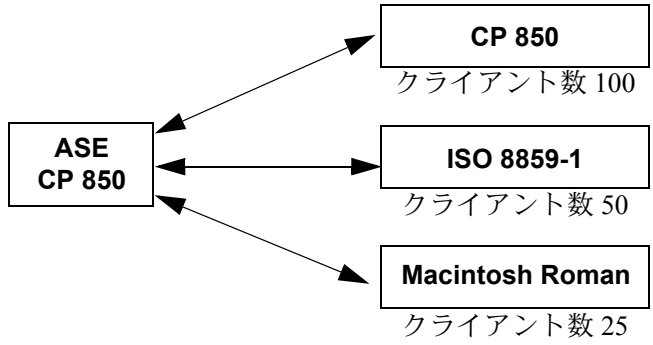
  クライアント・システムの文字セットとサーバのデフォルト文字セットが異なる場合は、データの整合性を保つためにデータを変換する必要があります。文字セット変換によるパフォーマンスの低下は大きくありませんが、できるだけ変換が少なくてすむデフォルト文字セットを選択するのが賢明です。

たとえば、ほとんどのクライアントが CP 850 を使用している場合は、サーバで CP 850 を指定します。これは、サーバを HP-UX システム (グループ 1 の言語のネイティブ文字セットが ROMAN8) にインストールする場合でも同じです。

注意 データベースを作成したり、Sybase 提供のデータベースに変更を加えたりする前に、デフォルトとして使用する文字セットを決定するよう強くおすすめします。

次の例 (図 9-2) では、175 のクライアントすべてが同じ Adaptive Server にアクセスします。これらのクライアントのプラットフォームも使用している文字セットも 1 つではありません。これらのクライアントが相互稼働するためには、クライアント／サーバ・システム内のすべての文字セットが同じ言語グループ (表 9-1 (288 ページ) を参照) に属している必要があります。Adaptive Server のデフォルト言語は CP 850 であり、最も多くのクライアントがこの文字セットを使用しています。したがって、文字セット変換の量が最小になり、サーバの動作効率が最大になります。

図 *9-2:* 同じ言語グループ内の異なる文字セットを使用するクライアント



サーバのデフォルト文字セットを選択するときには、一般的に使用されている文字セットをプラットフォームおよび言語別にまとめた次の表を参考にしてください。

表 *9-2:* 一般的な西欧クライアント・プラットフォーム

プラットフォーム	言語	文字セット
Win 95、98	米語、西欧語	CP 1252
Win NT 4.0	米語、西欧語	CP 1252
Win 2000	米語、西欧語	CP 1252
Sun Solaris	米語、西欧語	ISO 8859-1
HP-UX 10、11	米語、西欧語	ROMAN8
IBM AIX 4.x	米語、西欧語	ISO 8859-1

表 *9-3:* 一般的な日本語クライアント・プラットフォーム

プラットフォーム	言語	文字セット
Win 95、98	日本語	CP 932 for Windows
Win NT 4.0	日本語	CP 932 for Windows
Win 2000	日本語	CP 932 for Windows
Sun Solaris	日本語	EUC-JIS
HP-UX 10、11	日本語	EUC-JIS
IBM AIX 4.x	日本語	EUC-JIS

表 *9-4:* 一般的な中国語クライアント・プラットフォーム

プラットフォーム	言語	文字セット
Win 95、98	中国語 (簡体字)	CP 936 for Windows
Win NT 4.0	中国語 (簡体字)	CP 936 for Windows
Win 2000	中国語 (簡体字)	CP 936 for Windows
Sun Solaris	中国語 (簡体字)	EUC-GB
HP-UX 10、11	中国語 (簡体字)	EUC-GBS
IBM AIX 4.x	中国語 (簡体字)	EUC-GB

ソート順の選択

文字が同じでも言語によってソート順が異なります。たとえば、英語では *Cho* の位置が *Co* より前になりますが、スペイン語では逆になります。ドイツ語では、β は 1 つの文字ですが、辞書では二重文字 *ss* として扱われ、ソートされます。アクセント記号付き文字は特別な順序でソートされます。たとえば、*aménité* の位置は *améne* より前となり、アクセント記号がない場合とは逆になります。したがって、文字を正しくソートするためには、言語に固有のソート順が必要です。

各文字セットには、Adaptive Server がデータの並べ替えに使用するソート順が 1 つ以上定義されています。ソート順は、特定の言語または言語グループと特定の文字セットに関連付けられています。英語、フランス語、ドイツ語では、同じソート順を使用できます。これらの言語では、*A*、*a*、*B*、*b* のように、同じ文字は同じ順序でソートされるからです。特定の言語に固有の文字もあります。たとえば、アクセント記号付き文字 *é*、*à*、および *â* はフランス語では使用されますが、英語やドイツ語では使用されません。そのため、これらの文字のソート順に関して矛盾は発生しません。ただし、これはスペイン語には当てはまりません。二重文字である *ch* と *ll* のソート方法が異なります。したがって、これらの 4 言語すべてを同じ文字セットがサポートしますが、英語、フランス語、ドイツ語で使用されるソート順とスペイン語で使用されるソート順は異なります。

また、ソート順は特定の文字セットに関連付けられます。そのため、英語、フランス語、ドイツ語用のソート順は ISO 8859-1 に 1 セットあり、CP 850 にも 1 セットあります。特定の文字セットで使用できるソート順は、その文字セット・ディレクトリのソート順定義ファイル (**.srt*) に定義されています。文字セットと、その文字セットで利用できるソート順のリストについては、表 9-5 (298 ページ) を参照してください。

ソート順の使用

ソート順は以下のときに使用されます。

- インデックスの作成
- インデックス付きテーブルへのデータ格納
- order by 句の指定

ソート順の種類

すべての文字セットで、バイナリ・ソート順は必ず使用できます。このソート順では、文字セットの各文字に割り当てられたコード (「バイナリ」コード) の算術値だけに基づいてデータがソートされます。バイナリ・ソート順が適しているのは、各文字セットの最初の 128 文字 (ASCll 英語) とアジア系言語です。バイナリ・ソート順によって正しくソートされない可能性が最も高いのは、複数の言語をサポートする文字コード (グループ 1 や Unicode など) の場合です。このような文字セットを使用する場合は、別のソート順を選択してください。

文字セットによっては、以下に挙げる辞書ソート順も利用できます。

- 辞書順、大文字と小文字、およびアクセント記号を区別する − 大文字と小文字を異なる文字としてソートします。各種のアクセント記号付き文字を認識し、対応するアクセント記号なしの文字より後になるようにソートします。
- 辞書順、大文字と小文字を区別しない、アクセント記号を区別する − データを辞書順でソートしますが、大文字と小文字を異なる文字として認識しません。大文字と小文字は等しく処理され、ソート結果では両者が混在します。名前のテーブルでエントリの重複を避けたい場合に便利です。
- 辞書順、大文字と小文字を区別しない、アクセント記号を区別する、優先度を付けた順位 − 項目の同等性を評価する際に大文字と小文字の違いを認識しません。単語が大文字でも小文字でも同じ単語と見なされます。他の条件がすべて等しい場合は、大文字が優先されます (位置が先になる)。

  優先度付きで大文字と小文字を区別しないソート順を使用すると、order by 句で指定されたカラムがテーブルのクラスタード・インデックスのキーと一致した場合に、大きなテーブルでのパフォーマンスが低下することがあります。order by 句で、大文字と小文字の区別だけが異なる文字列を、大文字が先、小文字が後になるようにソートする必要がある場合を除いて、このソート順は選択しないでください。
- 辞書順、大文字と小文字、およびアクセント記号を区別しない − アクセント記号付き文字を、対応するアクセント記号なしの文字と同じものとして扱います。ソート結果では、アクセント記号付き文字と対応するアクセント記号なしの文字が混在します。

デフォルト・ソート順の選択

Sybase サーバでサポートされるデフォルト・ソート順は一度に 1 つだけです。すべてのユーザが同じ言語を使用している場合、またはユーザの言語がすべて同じソート順を使用する場合は、そのソート順を選択してください。たとえば、ユーザがフランス語のデータを扱っていて、フランス語のソート順を望んでいる場合は、フランス語の辞書ソート順からいずれかを選択します。ユーザが複数言語のデータを扱っていて、それらの言語が同じソート順を使用する場合は (たとえば、英語、フランス語、ドイツ語の場合など)、そのソート順を選択すれば、どの言語を使用するユーザにも対応できます。

ただし、ユーザが複数の言語を使用しており、それらの言語が異なるソート順を必要とする (たとえば、フランス語とスペイン語) 場合は、それらのソート順のどちらかをデフォルトとして選択しなければなりません。たとえば、フランス語のソート順を選択すると、スペイン語のユーザにとっては、二重文字の *ch* と *ll* が正しい順序でソートされません。インストール時には、デフォルトでサーバのソート順としてバイナリ・ソート順が設定されます。

sortkey 関数を使用して、言語ごとに代替ソート順を設定することができます。このソート順は、各ユーザが使用する言語に合わせて動的に選択できます。sortkey 関数はデフォルト・ソート順とは関連しておらず、両方が同じサーバに共存できます。sortkey 関数で設定するソート順は、範囲と詳細さの点でデフォルト・ソート順メカニズムが提供するソート順より優れています。詳細については、『リファレンス・マニュアル：ビルディング・ブロック』の「sortkey」と「compare」を参照してください。

表 *9-5:* 利用できるソート順

言語またはスクリプト	文字セット	ソート順
すべての言語	UTF-8	複数文字コードのソート順については、表 9-7 を参照
キリル： ブルガリア語、ベラルーシ語、マケドニア語、ロシア語、セルビア語、ウクライナ語	CP 855、CP 866、CP 1251、ISO 8859-5、Koi8、Macintosh Cyrillic	辞書順 (大文字と小文字、およびアクセント記号を区別する)
東欧： チェコ語、スロバキア語	CP 852、ISO 8859-2、CP 1250	辞書順 (大文字と小文字、およびアクセント記号を区別する) 辞書順 (大文字と小文字を区別しない、アクセント記号を区別する) 辞書順 (大文字と小文字、およびアクセント記号を区別する、優先度を付けた順位) 辞書順 (大文字と小文字、およびアクセント記号を区別しない)

言語またはスクリプト	文字セット	ソート順
英語、フランス語、ドイツ語	ASCII 8、CP 437、CP850、CP 860、CP 863、CP 1252a、ISO 8859-1、ISO 8859-15、Macintosh Roman、ROMAN8、ROMAN9、ISO 15	辞書順 (大文字と小文字、およびアクセント記号を区別する)
		辞書順 (大文字と小文字を区別しない、アクセント記号を区別する)
		辞書順 (大文字と小文字、およびアクセント記号を区別する、優先度を付けた順位)
		辞書順 (大文字と小文字、およびアクセント記号を区別しない)
英語、フランス語、ドイツ語	CP 850、CP 858	代替辞書順 (大文字と小文字を区別する)
		代替辞書順 (大文字と小文字を区別し、アクセント記号を区別しない)
		代替辞書順 (大文字と小文字を区別する、優先度を付けた順位)
ギリシャ語	ISO 8859-7	辞書順 (大文字と小文字、およびアクセント記号を区別する)
ハンガリー語	ISO 8859-2	辞書順 (大文字と小文字、およびアクセント記号を区別する)
		辞書順 (大文字と小文字を区別しない、アクセント記号を区別する)
		辞書順 (大文字と小文字、およびアクセント記号を区別しない)
日本語	EUCJIS、SJIS、DECKANJI	汎用 (大文字と小文字を区別しない、辞書順)
カザフ語	87	50
ロシア語	CP 866、CP 1251、ISO 8859-5、Koi8、Macintosh Cyrillic	辞書順 (大文字と小文字、およびアクセント記号を区別する)
		辞書順 (大文字と小文字を区別しない、アクセント記号を区別する)
スカンジナビア語	CP 850	辞書順 (大文字と小文字、およびアクセント記号を区別する)
		辞書順 (大文字と小文字を区別しない、優先度を付けた順位)
中国語 (簡体字)	EUC-GB、GB-18030、CP936	汎用 (大文字と小文字を区別しない、辞書順)
スペイン語	ASCII 8、CP 437、CP850、CP 860、CP 863、CP 1252、ISO 8859-1、ISO 8859-15、Macintosh Roman、ROMAN8	辞書順 (大文字と小文字、およびアクセント記号を区別する)
		辞書順 (大文字と小文字を区別しない、アクセント記号を区別する)
		辞書順 (大文字と小文字、およびアクセント記号を区別しない)
タイ語	CP 874、TIS 620	辞書順

言語またはスクリプト	文字セット	ソート順
トルコ語	ISO 8859-9	辞書順 (大文字と小文字、およびアクセント記号を区別する)
		辞書順 (大文字と小文字、およびアクセント記号を区別しない)
		辞書順 (大文字と小文字を区別しない、アクセント記号を区別する)
西欧語	CP 1252	辞書順 (大文字と小文字を区別しない、大文字と小文字を区別する、優先度順、アクセント記号を区別しない、スペイン語辞書、大文字と小文字を区別しないスペイン語、アクセントを区別しないスペイン語)

使用している言語がこの表にない場合は、その言語固有のソート順はありません。この場合は、バイナリ・ソート順を使用してください。また、sortkey 関数の使用でニーズが満たされるかどうかを調べてください。この表が示すように、多くの言語には複数のソート順があります。

中国語ピンイン・ソート順

ピンイン (かつての「漢語ピンイン」) では、ローマ字を使用して標準的な中国語の発音を表現します。ピンインは、標準中国語を読み書きする際に漢字を使わずに読み方をローマ字で表現したものです。ピンインでは、アクセント記号を使用して標準中国語の 4 音記号を表します。

以前のバージョンの Adaptive Server では、簡体字中国語 (GB) ソート順 (gbpinyin および gbpinyinocs) が使われており、Unilib 文字セットを使用した場合、GB 文字セットを使用しているデータベースのパフォーマンスが著しく低下していました。

Adaptive Server バージョン 15.0.3 では、自動的に gbpinyin および gbpinyinocs ソート順が使用されるため、処理ステップが省略されてパフォーマンスが大幅に向上します。

以前のバージョンの size of unilib cache 設定パラメータのデフォルト・サイズは 268KB でした。バージョン 15.0.3 では、デフォルト・サイズが 302KB に増えました。

ASCII および gbpinyin データにアクセスするクエリのパフォーマンスが向上しました。ただし、データ・セットに他の文字が混じっている場合は、パフォーマンスが改善しない場合があります。

Adaptive Server で gbpinyin および gbpinyinocs ソート順を使用するように設定する方法については、『システム管理ガイド』の「第 9 章 文字セット、ソート順、言語の設定」を参照してください。

中国語と日本語文字セットの大文字と小文字を区別しないソート順の選択

2 つのストアド・プロシージャを使用して、大文字と小文字を区別しないソート順を選択します。

- sp_helpsort
- sp_configure

sp_helpsort

sp_helpsort は、大文字と小文字を区別しない利用できるソート順を示します。

```
sp_helpsort
-----------
Name                      ID
--------------------------
nocase_eucgb              52
nocase_cp936              52
nocase_gb18030            52
nocase_eucjis             52
nocase_sjis               52
nocase_deckanji           52
```

sp_configure

大文字と小文字を区別しないソート順に切り替えるには、次のように入力します。

```
sp_configure 'default sortorder id', 52
```

デフォルト Unicode ソート順の選択

デフォルト Unicode ソート順は、サーバのデフォルト文字セットのソート順とは別のものです。この設定パラメータは静的パラメータであるため、変更した場合は、サーバの再起動と、unichar データのインデックス再構築が必要です。このソート順の識別には、一意性を保証するために、数値パラメータではなく文字列パラメータを使用します。

表 9-6 は、使用可能なデフォルトの Unicode ソート順を示します。

表 *9-6:* デフォルトの *Unicode* ソート順

名前	ID	説明
defaultml	20	デフォルト Unicode マルチ言語順
thaidict	21	タイ語辞書順
iso14651	22	ISO14651 標準順
utf8bin	24	UTF-8 バイナリと一致する UTF-16 順
binary	25	バイナリ・ソート
altnoacc	39	代替 (アクセント記号を区別しない)
altdict	45	代替 (辞書順)

名前	ID	説明
altnocsp	46	代替 (大文字と小文字を区別しない、優先度を付けた順位)
scandict	47	スカンジナビア語 (辞書順)
scannocp	48	スカンジナビア語 (大文字と小文字を区別しない、優先度を付けた順位)
bin_utf8	50	UTF-8 バイナリ・ソート順
dict	51	汎用 (辞書順)
nocase	52	汎用 (大文字と小文字を区別しない、辞書順)
nocasep	53	汎用 (大文字と小文字を区別しない、優先度を付けた順位)
noaccent	54	汎用 (アクセント記号を区別しない、辞書順)
espdict	55	スペイン語 (辞書順)
espnocs	56	スペイン語 (大文字と小文字を区別しない、辞書順)
espnoac	57	スペイン語 (アクセント記号を区別しない、辞書順)
rusnocs	59	ロシア語 (大文字と小文字を区別しない、辞書順)
cyrnocs	64	キリル語 (大文字と小文字を区別しない、辞書順)
elldict	65	ギリシャ語 (辞書順)
hundict	69	ハンガリー語 (辞書順)
hunnoac	70	ハンガリー語 (アクセント記号を区別しない、辞書順)
hunnocs	71	ハンガリー語 (大文字と小文字を区別しない、辞書順)
turknoac	73	トルコ語 (アクセント記号を区別しない、辞書順)

表 9-7 に、ロード可能なソート順を示します。

表 *9-7:* ロード可能なソート順

名前	ID	説明
cp932bin	129	CP932 のバイナリ順と一致する順
dynix	130	中国語 (簡体字) 順
gb3213bn	137	GB2312 のバイナリ順と一致する順
cyrdict	140	共通キリル語辞書順
turdict	155	トルコ語辞書順
euckscbn	161	EUCKSC のバイナリ順と一致する順
gbpinyin	163	中国語 (簡体字) 順
rusdict	165	ロシア語辞書順
sjisbin	179	SJIS のバイナリ順と一致する順
eucjisbn	192	EUCJIS のバイナリ順と一致する順
big5bin	194	BIG5 のバイナリ順と一致する順

Adaptive Server でこのソート順リストを表示するには、sp_helpsort を使用します。『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。

$SYBASE/collate/Unicode ディレクトリにある外部ファイルを使用して、ソート順を追加することができます。ソート順の名前と照合 ID は syscharsets に保存されます。外部 Unicode ソート順の名前が syscharsets に保存されていなくても、デフォルト Unicode ソート順を設定できます。

注意　外部 Unicode ソート順は Sybase から提供されます。外部 Unicode ソート順を新たに作成することはできません。

Unicode データに関連付けられるソート順は、従来の文字データに関連付けられるソート順とは完全に別のものです。Unicode データ型を使用する関係式はすべて、Unicode ソート順を使用して実行されます。これには、Unicode データと非 Unicode データの両方が関係する混合モードの式も含まれます。たとえば、次のクエリでは、varchar 型の文字定数 “Mü” が暗黙的に unichar にキャストされ、比較は Unicode ソート順に従って実行されます。

```
select * from authors where unicode_name > 'Mü'
```

他のすべての比較演算子、および連結演算子 “+”、演算子 “in”、演算子 “between” についても同様です。既に説明したように、目標は、既存のデータベース・アプリケーションとの互換性を保つことです。

等号によるテーブル・ジョイン (等価ジョイン) には、特に注意が必要です。このジョインは、通常、関連するカラムに定義されているインデックスを利用するように最適化されます。unichar 型カラムと char 型カラムとをジョインするときは、char 型カラムの変換が必要です。また、文字のソート順と Unicode のソート順は異なるので、オプティマイザは char 型カラムのインデックスを無視します。

Adaptive Server バージョン 12.5.1 以降では、サーバのデフォルト文字セットが UTF-8 に設定されていれば、サーバのデフォルト・ソート順 (char 型データのための) を前述のソート順のいずれかに設定できます。これより前のバージョンでは、UTF-8 に対して正常に機能するソート順はバイナリ・ソート順 “bin_utf8” (ID=50) だけでした。UTF-8 での char データのソート順を、unichar のソート順に対応するように選択することもできます。ただし、このことは必須ではありません。

Unicode のバイナリ・ソート順を選択するときは、注意が必要です。“binary” という名前のソート順は、unichar データ (UTF-16) のソート順として最も効率的であり、そのためデフォルトのソート順となっています。このソート順は、Unicode スカラ値に基づいています。つまり、32 ビットのサロゲート・ペアの順序は、16 ビットの Unicode 値の後になります。“utf8bin” というソート順は、UTF-8 の char データのデフォルトの (最も効率的な) バイナリ・ソート順 (“bin_utf8”) と一致するように設計されています。したがって、unichar 型に “binary”、UTF-8 char 型に “binary” を選択するか、unichar 型に “bin_utf8”、UTF-8 char 型に “bin_utf8” を選択することをおすすめします。最初の組み合わせは unichar 型のソート効率が高く、後者は char 型の効率が高くなります。UTF-8 char のソート順として “bin_utf8” を選択することは避けてください。このソート順は “bin_utf8” と同じですが、効率が悪いためです。

システム・メッセージ用言語の選択

Adaptive Server のインストール環境では、さまざまな言語のメッセージ・ファイルを備えた言語モジュールを使用できます。Adaptive Server には、英語、中国語 (簡体字)、フランス語、ドイツ語、日本語、韓国語、ポルトガル語 (ブラジル)、スペイン語のメッセージのための言語モジュールが用意されています。この他の言語をクライアントが使用する場合は、システム・メッセージはデフォルト言語の英語で表示されます。

同じサーバから出力されるメッセージの言語をクライアントごとに選択できます。たとえば、あるクライアントはフランス語、別のクライアントはスペイン語、さらに別のクライアントはドイツ語で、それぞれメッセージを表示することが可能です。ただし、そのためには、各クライアントが使用する言語がすべて同じ言語グループに属していなければなりません。たとえば、フランス語、スペイン語、ドイツ語はすべて言語グループ 1 に、一方、日本語は、日本語以外の言語を含んでいない言語グループ 101 に属します。したがって、サーバ言語が日本語である場合、システム・メッセージは日本語と英語のどちらかでしか表示できません。すべての言語グループがメッセージを英語で表示できます。サーバ全体のデフォルト言語も設定されており、ユーザが言語を選択していない場合は、そのデフォルト言語が使用されます。Unicode を選択した場合は、サポートされる言語であればどれでもシステム・メッセージを表示できます。

システム・メッセージの言語を選択するには、次の 2 つの方法があります。

- ユーザ・プロファイルにおいて言語を選択する
- *locales.dat* ファイルに言語を入力する

表 9-8 は、サポートされるシステム・メッセージ言語とその言語グループを示します。各ユーザは、セッションごとにシステム・メッセージ言語を 1 つだけ選択できます。

表 *9-8:* サポートされるシステム・メッセージ

言語グループ	システム・メッセージ言語	文字セット
グループ 1	フランス語、ドイツ語、スペイン語、ブラジルで使用するポルトガル語	ASCII 8、CP 437、CP 850、CP 860、CP 863、CP 1252、ISO 8859-1、ISO 8859-15、Macintosh Roman、ROMAN8
グループ 2	ポーランド語	CP 1250、CP 852、ISO 8859-2
グループ 101	日本語	CP 932、DEC Kanji、EUC-JIS、Shift-JIS
グループ 102	簡体字中国語 (PRC)	CP 936、EUC-GB、GB18030
グループ 104	韓国語	EUC-KSC、CP 949
グループ 105	タイ語	CP 874、TIS 620
Unicode	フランス語、ドイツ語、スペイン語、ブラジルで使用するポルトガル語、日本語、中国語 (簡体字)、韓国語	UTF-8
その他すべての言語グループ	英語	

クライアントでのメッセージの表示に使用されるすべての言語の言語モジュールをインストールしてください。言語モジュールのファイルは Adaptive Server インストール・ディレクトリの *locales* サブディレクトリにあり、「ローカライゼーション・ファイル」と呼ばれるファイル・グループの一部です。ローカライゼーション・ファイルとソフトウェア・メッセージ・ディレクトリ構造の詳細については、「ローカライゼーション・ファイルの種類」(320 ページ)を参照してください。

サーバの設定：例

この項では、設定オプションとその設定手順について説明します。以下の手順はあくまでも例であり、実際の設定作業に役立つよう概念と方法を例示したものです。

スペイン語版サーバ

この例では、すべてのクライアントが同じ言語を使用する場合に新しいサーバを設定する方法について説明します。必要な作業は次のとおりです。

1　サーバ言語を選択します。この例ではスペイン語です。表 9-1 (288 ページ)に示すように、スペイン語は言語グループ 1 に属しています。使用するプラットフォームに基づいて、言語グループ 1 の文字セットを選択します。最も多くのクライアントが使用する文字セットを選択するようおすすめします。将来、他の国にも事業を拡大し、各国の言語に対応する必要があると思われる場合は、Unicode のインストールを検討してください (「サーバの文字セットの選択」(287 ページ) を参照)。

2　サーバにスペイン語の言語モジュールをインストールします。これによって、クライアントにはシステム・メッセージがスペイン語で表示されます。

3　デフォルト・ソート順を選択します。表 9-5 (298 ページ) に示すように、スペイン語にはバイナリ・ソート順の他に 3 つのソート順があります。その中からソート順を 1 つ選択します。

4　サーバを再起動します。

アメリカ企業の日本法人

この例では、クライアントは日本にあり、データの入力およびソートとシステム・メッセージの受信には日本語を使用しますが、データの送信先となるサーバは英語だけを理解するユーザがアクセスします。

1 サーバのデフォルト文字セットを選択します。言語グループ 101 (日本語) の文字セットをインストールした場合は、同じサーバで日本語と英語の両方のデータをサポートできます。

2 システム・メッセージを日本語で表示するために、日本語の言語モジュールをインストールします。

3 ソート順を選択します。表 9-5 (298 ページ) に示すように、日本語にはバイナリ・ソート順しかありません。したがって、英語と日本語のどちらのクライアントでもバイナリ・ソート順がデフォルト・ソート順となります。両クライアントのユーザに対する解決策として sortkey 関数の使用を検討してください。

4 日本人ユーザがデフォルトで日本語のメッセージを要求するように設定されていることを確認します。言語グループ 101 の文字セットを使用するのに加え、日本語の言語モジュールをインストールしているので、日本のクライアントにはメッセージが日本語で表示され、アメリカのクライアントはメッセージ言語として英語または日本語を選択できます。

クライアントが複数の国にある日本企業

この会社は日本にあり、フランス、ドイツ、スペインにクライアントがあります。このケースでは、同じサーバにヨーロッパ系の言語とアジア系の言語を併存させる必要があります。

1 デフォルトのサーバ言語と文字セットを選択します。この会社の本社は日本にあり、クライアントのほとんどが日本にあるため、デフォルトのサーバ言語は日本語でなければなりません。しかし、フランス、ドイツ、スペインのクライアントがデータをそれぞれのネイティブ言語で送受信できるようにすることも必要です。表 9-1 (288 ページ) に示すように、日本語は言語グループ 101 に属しているのに対し、フランス語、ドイツ語、スペイン語は言語グループ 1 に属しています。必要な言語がすべて同じ言語グループに属してはいないので、これらの言語を同じサーバでサポートする唯一の方法は、デフォルト文字セットとして Unicode を選択することです。

2 日本語、フランス語、ドイツ語、スペイン語の言語モジュールをインストールします。

3 バイナリ・ソート順を選択します。これが Unicode 文字セットで使用できる唯一のソート順だからです (ただし、各ユーザの好みに従ってデータをソートできるように、アプリケーション・コードに sortkey 関数を追加することも検討します)。

4　システム・メッセージのデフォルト言語として日本語を選択します。他の国のクライアントでは、それぞれのネイティブ言語をメッセージ言語として選択できます。

文字セット、ソート順、メッセージ言語の変更

サーバの設定が完了した後でも、システム管理者は Adaptive Server のデフォルト文字セット、ソート順、メッセージ言語を変更できます。ソート順は特定の文字セットに基づいて作成されるので、文字セットを変更すると通常はソート順も変更されます。しかし、1 つの文字セットに対して利用できるソート順は複数あるので、文字セットを変更しなくてもソート順を変更できます。

Adaptive Server のデフォルトのソート順、文字セット、プライマリ・ソート順のテーブルを表示するには、次のように入力します。

```
sp_helpsort
```

デフォルト文字セットの変更

Adaptive Server の「デフォルト文字セット」は 1 つだけ指定できます。この文字セットを使用してデータベース内にデータが保管されます。Adaptive Server をインストールするときに、デフォルト文字セットを指定します。

警告！ Adaptive Server のデフォルト文字セットを変更する場合は、次の指示に従い、十分に注意して実行してください。デフォルト文字セットを変更する前にバックアップを行うよう強くおすすめします。

Adaptive Server のデフォルト文字セットを変更するときは、既存のデータを新しいデフォルト文字セットに変換する必要があります。変換が不要なのは以下の場合だけです。

- サーバ内にユーザ・データがない。
- サーバ内のユーザ・データが破壊されてもかまわない。
- サーバ内のデータが ASCll-7 だけを使用しているという絶対の確信がある。このような場合は、デフォルトを変更する前にデータをサーバからコピー・アウトする必要はありません。

それ以外の場合は、以下の手順で必ず既存のデータを変換してください。

1　bcp を使用してデータをコピー・アウトします。

2　デフォルト文字セットを変更します。

3　データ変換用のフラグを指定して bcp を実行し、データをサーバにコピー・インします。

bcp を使用したデータのコピー方法の詳細については、『ユーティリティ・ガイド』を参照してください。

警告！ データを別の文字セット (特に UTF-8) に変換すると、変換後のデータが大きくなりすぎ、割り当てられたカラム・サイズに入り切らなくなることがあります。データが入りきらないカラムは作成し直す必要があります。

既存のデータの文字セットと新しいデフォルト文字セットとの間のコード変換がサポートされている必要があります。サポートされていないと、変換エラーが発生して、データは正しく変換されません。サポートされている文字セット変換の詳細については、「第 10 章 クライアント／サーバの文字セット変換の設定」を参照してください。

文字セット間で変換がサポートされていても、文字セット間に多少相異があることや、他の文字セットに相当する文字がないなどの理由から、エラーが発生することがあります。不完全な文字や無効な文字のあるデータのローを、データベースにコピー・インしないでください。

リソース・ファイルを使ったソート順の変更

リソース・ファイルを使用して、Adaptive Server の文字セットを変更できます。サンプル・リソース・ファイル *sqlloc.rs* は、*$SYBASE/ASE-12_5/init/sample_resource_files/* にあります。

Adaptive Server 12.5.1 インストール環境のリソース・ファイルは次のようになります。

```
sybinit.release_directory:USE_DEFAULT
sqlsrv.server_name:PUT_YOUR_SERVER_NAME_HERE
sqlsrv.sa_login:sa
sqlsrv.sa_password:
sqlsrv.default_language:USE_DEFAULT
sqlsrv.language_install_list:USE_DEFAULT
sqlsrv.language_remove_list:USE_DEFAULT
sqlsrv.default_characterset:USE_DEFAULT
sqlsrv.characterset_install_list:USE_DEFAULT
sqlsrv.characterset_remove_list:USE_DEFAULT
sqlsrv.sort_order:USE_DEFAULT
# An example sqlloc resource file...
# sybinit.release_directory:USE_DEFAULT
# sqlsrv.server_name:PUT_YOUR_SERVER_NAME_HERE
# sqlsrv.sa_login:sa
# sqlsrv.sa_password:
# sqlsrv.default_language:french
# sqlsrv.language_install_list:spanish,german
# sqlsrv.language_remove_list:USE_DEFAULT
# sqlsrv.default_characterset:cp437
# sqlsrv.characterset_install_list:mac,cp850
# sqlsrv.characterset_remove_list:USE_DEFAULT
# sqlsrv.sort_order:dictionary
```

デフォルト・ソート順の変更

Adaptive Server の「デフォルトのソート順」は 1 つだけ設定できます。これはデータを並べ替えるために使用される照合順序です。特定の Adaptive Server の文字データのソート順を変更するときは、次の点に注意してください。同じ組織に複数の Adaptive Server がある場合は、すべて同じソート順を使用することをおすすめします。ソート順を 1 つにすることで一貫性が保たれ、分散処理の管理が簡単になります。

デフォルト・ソート順を変更すると、インデックスの再構築が必要になる場合があります。詳細については、「文字セット、ソート順、メッセージ言語の再設定」(309 ページ) を参照してください。

文字セット、ソート順、メッセージ言語の再設定

この項では、Adaptive Server のデフォルトの文字セット、ソート順、メッセージ言語を変更する前と後に行う手順の要約を説明します。新しいサーバの文字セット、ソート順、メッセージ言語を設定する方法については、プラットフォームの『Adaptive Server Enterprise 設定ガイド』を参照してください。

文字セットまたはソート順を変更する前と後に、Adaptive Server のすべてのデータベースのバックアップを作成してください。次の条件が当てはまる場合は、データベースをバックアップした後に bcp を使用してデータをコピー・インおよびコピー・アウトしてください。

- データベース内に文字データがあり、そのデータを新しい文字セットに変換する必要がある。このような場合は、サーバのデフォルト文字セットの変更後にデータのデータベース・ダンプをロードしないでください。ロードされるデータは新しい文字セットに基づくものと解釈され、データは破損した状態になります。
- 変更するのはデフォルト・ソート順だけで、デフォルト文字セットは変更しない。このような場合は、ソート順を変更する前に作成したダンプからデータベースをロードすることはできません。ロードしようとすると、エラー・メッセージが表示され、ロードはアボートされます。
- デフォルトの文字セットを変更するが、新旧どちらかのソート順がバイナリでない。このような場合は、文字セットを変更する前に作成したデータベース・ダンプをロードすることはできません。

新旧どちらの文字セットもバイナリ・ソート順を使用し、両方の文字セット間の変換を必要としない場合を除き、デフォルト文字セットとソート順を再設定した後にデータベース・ダンプからデータを再ロードすることはできません。詳細については、「デフォルト文字セットの変更」(307 ページ) を参照してください。

Unicode の例

次の例では、xpubs という名前の架空のデータベースで univarchar 型のカラムを使用するように変更します。

スキーマ

インストール時にすべてデフォルトを選択して設定されたサーバ (文字セットは “iso_1”、デフォルト・ソート順は ID 50 の “binary_iso_1”) 上で、次のスクリプトを使用してデータベースを作成したとします。

```
> create database xpubs
> go
> use xpubs
> go
> create table authors (au_id int, au_lname varchar(255),
au_fname varchar(255))
> go
> create index au_idx on authors(au_lname, au_fname)
> go
```

その後で、一連の挿入と更新によって、データがサーバにロードされました。

UTF-8 への変換

Unicode を使用するには、初めに、データを抽出して UTF-8 形式に変換します。

```
% bcp xpubs..authors out authors.utf8.bcp -c -Jutf8 -Usa -P
```

次に、UTF-8 をデフォルト文字セットとしてサーバにインストールします。

```
% charset -Usa -P binary.srt utf8
% isql -Usa -P
> sp_configure 'default sortorder id', 50, 'utf8'
> go
> shutdown
> go
```

サーバを再起動すると、デフォルト文字セットが変更され、システム・テーブルのインデックスが再作成されます。もう一度サーバを再起動して、データを再ロードします。

```
% isql -Usa -P
> sp_dboption xpubs, 'select into', true
> go
> use xpubs
> go
> checkpoint
> go
> delete from authors
> go
> quit

% bcp xpubs..authors in authors.utf8.bcp -c -Jutf8 -Usa -P
```

選択したカラムを unichar にマイグレートする

作業データベースのデフォルト文字セットが UTF-8 に設定されている場合は、選択したカラムを univarchar に簡単に変換できます。

```
% isql -Usa -P
> use xpubs
> go
> alter table authors modify au_lname univarchar(255),
au_fname univarchar(255)
> go
```

カラムが新しいデータ型に変更され、データが適切に変換され、インデックスが再作成されます。

unitext へ、または unitext からのマイグレーション

現時点では、alter table modify コマンドには、text、image、または unitext カラムを指定できません。text カラムを unitext カラムにマイグレートするには、bcp を実行し、unitext カラムのテーブルを作成し、再度 bcp を実行して新しいテーブルにデータを挿入します。この方法でマイグレートするには、bcp の呼び出し時に -Jutf8 オプションを指定する必要があります。

準備手順

インストール・プログラムを実行して Adaptive Server を再設定する前に、次の手順を実行してください。

1　すべてのユーザ・データベースと master データベースをダンプします。model または sybsystemprocs の内容を変更した場合は、これらもダンプします。

2　言語モジュールがロードされていない場合はロードしてください (詳細は、プラットフォームの『Adaptive Server Enterprise 設定ガイド』を参照してください)。

3　Adaptive Server のデフォルト文字セットを変更する場合で、現在使用しているデータベースに ASCII-7 以外のデータが含まれているときは、bcp を使用してデータベースの既存のデータをコピー・アウトしてください。

言語モジュールをロードした後で、Adaptive Server のインストール・プログラムを実行して以下の操作を行うことができます。

- Adaptive Server に組み込まれているメッセージ言語と文字セットをインストールまたは削除する。
- デフォルトのメッセージ言語または文字セットを変更する。
- 別のソート順を選択する。

インストール・プログラムの使用方法については、プラットフォームの『Adaptive Server Enterprise 設定ガイド』を参照してください。

注意 文字セットやソート順を変更するには、サーバで管理されているデータベースをすべてオープンしておく必要があります。オープンしているデータベースの数が不足していると、ソート順を変更してから再起動したときに、次のメッセージがエラー・ログに記録され、サーバのソート順は以前のソート順に戻ります。

```
The configuration parameter 'number of open databases' must be
at least as large as the number of databases, in order to change
the character set or sort order."Re-start Adaptive Server, use
sp_configure to increase 'number of open databases' to at least
%d, then re-configure the character set or sort order
```

言語、文字セット、またはソート順を再設定する必要がある場合は、『ユーティリティ・ガイド』で説明している sqlloc ユーティリティを使用してください。Windows を使用している場合は、『設定ガイド』の「第 3 章 Adaptive Server のデフォルト設定」で説明している Server Config ユーティリティを使用してください。

追加の言語をインストールしたけれども、Adaptive Server の文字セットやソート順は変更していない場合は、以上で再設定のプロセスは終了です。

Adaptive Server のデフォルト文字セットを変更した場合で、現在のデータベースに ASCII-7 以外のデータがあるときは、bcp を使用してデータをデータベースにコピー・インします。このとき、変換のためのフラグを指定してください。

Adaptive Server のデフォルトのソート順または文字セットを変更した場合は、「文字セット、ソート順、メッセージ言語の再設定」(309 ページ) を参照してください。

ユーザのデフォルト言語の設定

追加の言語がインストールされると、クライアント・プログラムを実行するユーザは、sp_modifylogin を実行してその言語をデフォルト言語として設定することや、locales.dat に定義されているエントリを使用してクライアント・マシンの LANG 変数を設定することができます。

再設定後のリカバリ

Adaptive Server が停止して再起動するたびに、各データベースのリカバリが自動的に実行されます。自動リカバリについては、『システム管理者ガイド 第 2 巻』の「第 11 章 バックアップおよびリカバリ・プランの作成」を参照してください。

リカバリが完了すると、新しいソート順と文字セットの定義がロードされます。

ソート順を変更すると Adaptive Server はシングルユーザ・モードに切り替わります。このとき、システム・テーブルに対して必要な更新を加えることができますが、他のユーザはサーバを使用できなくなります。文字ベースのインデックスを持つシステム・テーブルについては、ソート順の変更によって壊れたインデックスがあるかどうかのチェックが自動的に行われます。テーブルの文字ベースのインデックスは、必要に応じて新しいソート順の定義を使用して自動的に再構築されます。

システムのインデックスが再構築された後で、文字ベースのユーザ・インデックスについて、sysindexes システム・テーブル内に「suspect (疑わしい)」というマークが付けられます。疑わしいインデックスを持つユーザ・テーブルについては、sysobjects テーブル内で 「read only (読み込み専用」というマークが付けられます。これによって、疑わしいインデックスのチェックと、必要に応じて再構築が実行されるまでは、このようなテーブルへの更新とインデックスの使用はできなくなります。

範囲分割されたユーザ・テーブルの文字ベースのパーティション・キーがチェックされ、ソート順の変更または文字セットの変更によってパーティションが破損している可能性がある場合は、「suspect (疑わしい」のマークが付きます。

次に、設定情報を保持するディスク領域にある古い情報が、新しいソート順情報に置き換えられます。その後、Adaptive Server が停止します。次のセッションを開始するときは、システム情報は完全で正確な状態になっています。

sp_indsuspect による壊れたインデックスの検索

Adaptive Server の停止後に再起動して sp_indsuspect を実行すると、どのユーザ・テーブルのインデックス再構築が必要かを調べることができます。

```
sp_indsuspect [tab_name]
```

この場合、*tab_name* はチェックするテーブルの名前です。*tab_name* を指定しないで sp_indsuspect を実行すると、現在のデータベースにあるテーブルのうち、ソート順の変更時に「suspect (疑わしい」というマークが付けられたインデックスを持つすべてのテーブルのリストが作成されます。

次の例は、mydb データベースで sp_indsuspect を実行した結果、疑わしいインデックスが 1 つ見つかったことを示しています。

```
sp_indsuspect

Suspect indexes in database mydb
Own.Tab.Ind (Obj_ID, Ind_ID) =
dbo.holdings.h_name_ix(160048003, 2)
```

ソート順を変更した後のインデックスの再構築

dbcc reindex によって、dbcc checktable の「高速」バージョンが実行され、ユーザ・テーブルのインデックスの整合性が検査されます。詳細については、『システム管理ガイド 第 2 巻』の「第 10 章 データベースの一貫性の検査」を参照してください。dbcc reindex は、使用されているソート順が新しいソート順と一致しないインデックスを削除して、再構築します。dbcc reindex が最初のインデックス関連エラーを発見すると、メッセージが表示され、ソート順の違うインデックスは再構築されます。Adaptive Server でのソート順を変更した後に、システム管理者またはテーブル所有者は dbcc reindex を実行してください。

```
dbcc reindex ({table_name | table_id})
```

sp_indsuspect によって表示された、疑わしいインデックスを持つすべてのテーブルに対してこのコマンドを実行してください。次に例を示します。

```
dbcc reindex(titles)

One or more indexes are corrupt. They will be rebuilt.
```

これは、dbcc reindex で、titles テーブルに 1 つまたは複数の疑わしいインデックスが発見されたことを示しています。dbcc reindex は、該当するインデックスに対して削除と再作成を実行します。

テーブルのインデックスが正しい場合やテーブルにインデックスがない場合は、dbcc reindex を実行したときにインデックスの再構築は行われませんが、代わりに、メッセージが表示されます。テーブルに破壊されたデータが含まれているという疑いがある場合は、このコマンドはアボートされます。この場合、dbcc checktable の実行を指示するエラー・メッセージが表示されます。

dbcc reindex が正常終了したときは、そのテーブルのインデックスの「suspect (疑わしい)」マークはすべて削除されています。テーブルの「read-only (読み込み専用)」マークも削除され、テーブルを更新できるようになります。インデックスを再構築する必要があるかどうかに関係なく、この 2 つのマークは削除されます。

dbcc reindex を使用して、システム・テーブルのインデックスを再構築することはできません。システム・インデックスは、Adaptive Server がソート順を変更して再起動した後で、自動リカバリの一部として必要に応じて検査され再構築されます。

文字セットを変更した後の *text* データのアップグレード

Adaptive Server の文字セットをマルチバイト文字セットに変更した場合は、dbcc fix_text を使用して text 型の値をアップグレードしてください。

text 型の値は複数のページにわたるほどの大きさになることがあるため、Adaptive Server はページ境界を越える文字データを処理する必要があります。これには、個々の text ページに関する追加の情報が必要です。システム管理者またはテーブル所有者は、text データが含まれる各テーブルについて、dbcc fix_text を実行して、必要な新しい値を計算する必要があります。

text データが含まれているすべてのテーブルの名前を調べるには、次のクエリを使用してください。

```
select sysobjects.name
from sysobjects, syscolumns
where syscolumns.type = 35
and sysobjects.id = syscolumns.id
```

システム管理者またはテーブル所有者は、必要となる新しい値を計算するために、dbcc fix_text を実行してください。

dbcc fix_text の構文は次のとおりです。

dbcc fix_text (*table_name* | *table_id*)

指定するテーブルは、現在のデータベースになければなりません。

dbcc fix_text は、指定されたテーブルをオープンし、text 値ごとに必要な文字統計値を計算し、適切なページ・ヘッダ・フィールドに追加します。この処理は、テーブルにある text 値のサイズによっては時間がかかることがあります。dbcc fix_text によって多数のログ・レコードが生成され、トランザクション・ログが満杯になることがあります。dbcc fix_text は一連の小さなトランザクションで更新を行うので、ログが満杯になっても、失われる作業はわずかです。

ログ・スペースが不足した場合は、ログを消去してください。詳細については、『システム管理ガイド 第 2 巻』の「第 12 章 ユーザ・データベースのバックアップとリストア」を参照してください。その後、元の dbcc fix_text が停止したときに更新していたのと同じテーブルを指定して、dbcc fix_text を再起動してください。マルチバイトの text 値のそれぞれに、その text 値がアップグレードされたかどうかを示す情報が含まれています。したがって、dbcc fix_text は、前回までに処理されていない text 値だけをアップグレードします。

データベースのログが別のセグメントに保管されている場合は、スレッショルドを使用してログのクリアを管理できます。『システム管理ガイド 第 2 巻』の「第 16 章 スレッショルドによる空き領域の管理」を参照してください。

dbcc fix_text がテキスト・ページに対する必要なロックを取得できない場合は、次のように問題点がレポートされ、処理は続行します。

```
Unable to acquire an exclusive lock on text page 408. This text
value has not been recalculated.  In order to recalculate those
TEXT pages you must release the lock and reissue the dbcc fix_text
command.
```

文字セットを変更した後の *text* 値の検索

マルチバイトの文字セットに変更した後で text 値を検索しようとしたとき、dbcc fix_text をまだ実行していない場合は、コマンドは正しく実行されず、次のエラー・メッセージが生成されます。

```
Adaptive Server is now running a multi-byte character set, and
this TEXT column's character counts have not been recalculated
using this character set. Use dbcc fix_text before running this
query again.
```

ソート順または文字セットを変更した後でエラーが発生した場合は、『ASE トラブルシューティング＆エラー・メッセージ・ガイド』の「ソート順またはデフォルトの文字セットを手動で変更する方法」を参照してください。

疑わしいパーティションの処理

パーティションに「suspect (疑わしい)」のマークが付けられる理由は次の 2 つです。

- 範囲分割テーブルのソート順または文字セットが変更された
- ハッシュ分割されたテーブルについてプラットフォーム間でダンプとロードが行われた

テーブルに疑わしいパーティションがあるとマークされた場合、次の処理が行われます。

- このテーブルのすべての更新およびカーソル・アクティビティが保留されます。
- partition by を除き、alter table コマンドは使用できません。疑わしいパーティションのあるテーブルについて、create index および drop index は使用できません。
- 疑わしいパーティションのあるテーブルに対して、select コマンドを使用できます。ただし、オプティマイザは、破損の可能性のあるパーティション条件の使用を回避するために、このようなテーブルをラウンドロビン方式で分割されたテーブルとして処理します。

疑わしいパーティションを含んでいるテーブルの修正

- ソート順の変更後にパーティション条件の修正が必要な場合は、alter table コマンドと一緒に partition by オプションを使用して疑わしいパーティションのあるテーブルを再分割してください。
- パーティション条件の修正が必要ない場合は、reorg rebuild table コマンドを使用してテーブルを再構築し、パーティション間にデータ・ローだけを再配分します。
- テーブルのパーティションだけでなくインデックスも「疑わしい」とされた場合は、partition by または reorg rebuild を使用して疑わしいインデックスとパーティションの両方を修正してください。

プラットフォーム間のダンプとロード操作時の疑わしいパーティションの処理

- 初めての online database コマンドの実行中にエンディアン・タイプの異なる 2 つのプラットフォームで load database を実行すると、ハッシュ分割は “suspect (疑わしい)” のマークが付けられます。
- unichar または varchar 分割キーで内部生成されたパーティション条件を持つラウンドロビン分割のグローバル・クラスタード・インデックスは、“suspect (疑わしい)” のマークが付けられます。
- データベースがオンラインになったら、sp_post_xpload を使用して疑わしい分割およびインデックスを修正します。

サポートされていない言語の日付文字列のインストール

sp_addlanguage を使用すると、言語モジュールが用意されていない言語での曜日と月の名前をインストールできます。sp_addlanguage を使用して、次のものを定義します。

- 言語の名前とその名前のエイリアス (オプション)
- 月のフルネームのリストと省略名のリスト
- 曜日のフルネームのリスト
- 日付の入力フォーマット (たとえば、月／日／年)
- 最初の曜日の番号

次にイタリア語の情報を追加する例を示します。

```
sp_addlanguage italian, italiano,
"gennaio,febbraio,marzo,aprile,maggio,giugno,luglio,agosto,settembre,ottobre,
novembre,dicembre",
"genn,feb,mar,apr,mag,giu,lug,ago,sett,ott,nov,dic",
"lunedi,martedi,mercoledi,giovedi,venerdi,sabato,domenica",
dmy, 1
```

sp_addlanguage では、正確なデータの入力規則が適用されます。月の名前、月の省略名、曜日のリストはカンマで区切ります。途中にスペースや行送り (改行) を入れないでください。また、指定する要素の数が正確でなければなりません (月の文字列は 12 個、曜日の文字列は 7 個です)。

日付フォーマットの有効値は、mdy、dmy、ymd、ydm、myd、dym です。dmy と指定すると、日付の順は「日／月／年」となります。このフォーマットは、データの入力だけに影響します。出力フォーマットを変更するときは、convert 関数を使用してください。

サーバとクライアントでの日付の解釈

通常、日付値はクライアント側で分解されます。ユーザが日付値を選択すると、Adaptive Server はその値を内部フォーマットでクライアントに送ります。クライアントは、そのクライアント上の *locales* ディレクトリのデフォルト言語サブディレクトリにある *common.loc* ファイルと他のローカライゼーション・ファイルを使用して、内部フォーマットを文字データに変換します。たとえば、ユーザのデフォルト言語がスペイン語の場合、Adaptive Server は */locales/spanish/char_set* ディレクトリにある *common.loc* ファイルを探します。このファイルの情報を使用して、たとえば 12 febrero 2007 と表示します。

Adaptive Server の言語モジュールが提供されていないイタリア語がユーザのデフォルト言語であり、イタリア語の日付値が追加されているとします。クライアントは、サーバに接続するときにイタリア語の *common.loc* ファイルを探しますが、そのファイルは見つかりません。クライアントはエラー・メッセージを表示し、そのサーバに接続します。その後、ユーザが日付値を選択すると、日付は米語フォーマットで表示されます。sp_addlanguage で追加された日付値を表示するには、convert 関数を使用して、サーバで日付を文字データに強制的に変換してください。

次のクエリで生成される結果セットでは、日付は米語フォーマットです。

```
select pubdate from titles
```

次のクエリの場合は、月の名前がイタリア語で返されます。

```
select convert(char(19),pubdate) from titles
```

国際化ファイルとローカライゼーション・ファイル

国際化ファイルの種類

特定の言語でデータ処理をサポートするためのファイルを「国際化ファイル」と呼びます。Adaptive Server に付属する国際化ファイルには、いくつかの種類があります。

表 *9-9:* 国際化ファイル

ファイル	ロケーション	目的と内容
charset.loc	*charsets* ディレクトリにある各文字セットのサブディレクトリ	英数字、句読点、オペランド、大文字、小文字などの文字の辞書的属性を定義する、文字セット定義ファイル。Adaptive Server がデータを正しく処理するために必要。
**.srt*	*charsets* ディレクトリにある各文字セットのサブディレクトリ	文字、数字、特殊文字のソート順を定義する。合字や発音区別符号などの言語特有の規則も考慮される。
**.xlt*	*charsets* ディレクトリにある各文字セットのサブディレクトリ	bcp や isql などのユーティリティで使用される、端末固有の文字を変換するファイル。*.xlt* ファイルの使用方法の詳細については、「第 10 章 クライアント／サーバの文字セット変換の設定」と『ユーティリティ・ガイド』を参照してください。

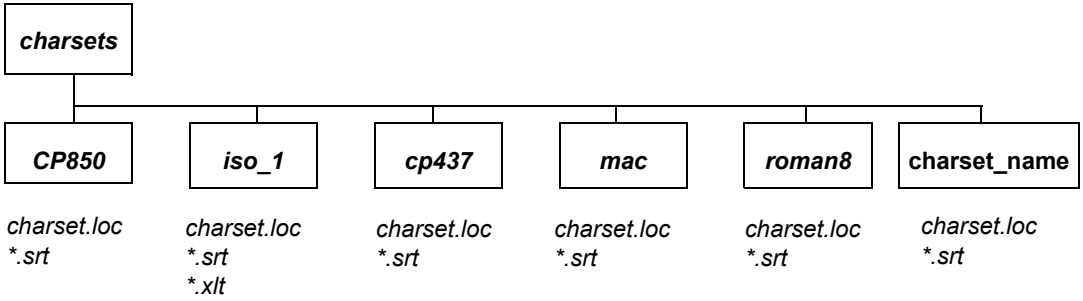
警告！ 国際化ファイルは変更できません。新しい端末定義またはソート順のインストールが必要な場合は、最寄りの Sybase 社または販売代理店に連絡してください。

文字セットのディレクトリ構造

図 9-3 は、Adaptive Server とともに提供される西欧言語の文字セットのディレクトリ構造を示します。*charsets* ディレクトリには、文字セットごとに別のサブディレクトリがあります。それぞれの文字セット (*cp850* など) のサブディレクトリには、文字セットとソート順の定義ファイルおよび端末固有のファイルがあります。

追加の文字セットをロードした場合も、その文字セットのサブディレクトリが *charsets* ディレクトリの下に作成されます。

図 9-3: charsets ディレクトリの構造



次のグローバル変数には、文字セットの情報が格納されています。

表 9-10: 文字セットのグローバル変数

グローバル変数	説明
@@char_convert	文字セット変換が有効でない場合は 0 が格納される。文字セット変換が有効な場合は 1 が格納される。
@@client_csname	クライアントの文字セット名。クライアント文字セットが初期化されていない場合は NULL に設定され、初期化されている場合は、その接続の文字セット名が格納される。
@@client_csid	クライアント文字セットが一度も初期化されていない場合、-1 に設定される。一度でも初期化された場合、syscharsets から返された接続のためのクライアント文字セット ID に設定される。
@@client_csexpansion	サーバの文字セットをクライアントの文字セットに変換するときに使用する拡張係数を返す。
@@maxcharlen	Adaptive Server のデフォルト文字セット中の 1 文字の最大長 (バイト)。
@@ncharsize	現在のサーバのデフォルト文字セット中の 1 文字の最大長 (バイト)。
@@unicharsize	2 に等しい。

ローカライゼーション・ファイルの種類

Adaptive Server の言語モジュールごとに、表 9-11 に示すローカライゼーション・ファイルが用意されています。

表 9-11: ローカライゼーション・ファイル

ファイル	ロケーション	目的と内容
locales.dat	*locales* ディレクトリ内	デフォルト・メッセージ言語とデフォルト文字セットを識別するためにクライアント・アプリケーションが使用する。
server.loc	*$SYBASE/$SYBASE_ASE/locales* ディレクトリ内の言語サブディレクトリの下の文字セット・サブディレクトリ内	ローカル言語に翻訳されたソフトウェア・メッセージ。Sybase 製品は製品固有の **.loc* ファイルを持つ。内容が翻訳されていない場合は、そのソフトウェアのメッセージまたはテキストはローカル言語ではなく米語で表示される。
common.loc	*locales* ディレクトリの言語ディレクトリと文字セットのディレクトリ内	各国の言語での月名とその短縮形、および各国で使われる日付、時間、通貨のフォーマットに関する情報。

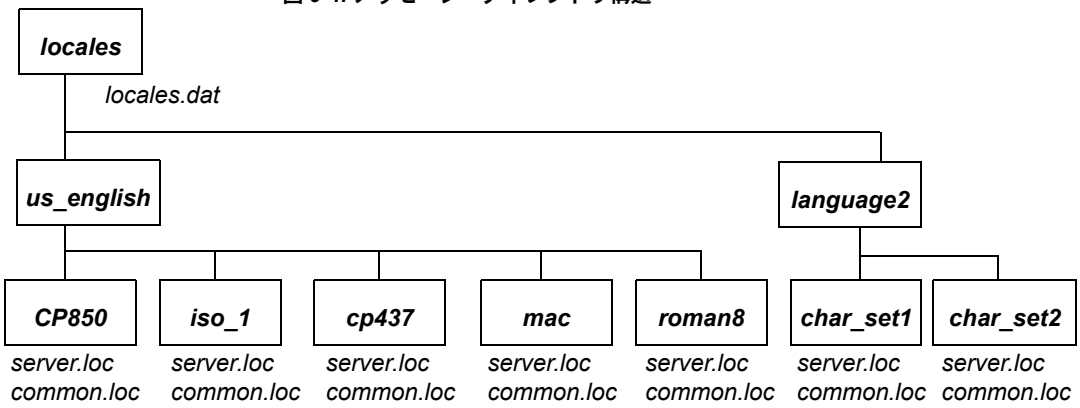
dataserver、sqlloc、syconfig などが使用するすべての Adaptive Server 関連ロケール・ファイルは *$SYBASE/SYBASE_ASE/locales* にあります。すべての Open Client/Server 関連ロケール・ファイル (ctlib、ctisql、ctbcp、optdiag、installjava など) は *$SYBASE/locales* にあります。

警告！ ローカライゼーション・ファイルは変更できません。このファイル内の情報を変更する必要がある場合は、最寄りの Sybase 社または販売代理店に連絡してください。

ソフトウェア・メッセージのディレクトリ構造

図 9-4 は、ローカライゼーション・ファイルがどのように配置されているかを示します。*locales* ディレクトリ内には、インストールされた言語ごとにサブディレクトリがあります。*us_english* サブディレクトリは必ずあります (PC プラットフォームでは、このディレクトリの名前は *english* です)。インストール中に、Adaptive Server 上にインストールする言語を選択するときに、サポートされているソフトウェア・メッセージ言語のリストが画面に表示されます。言語を追加するための言語モジュールをインストールすると、その言語のサブディレクトリが作成されます。それぞれの言語のサブディレクトリには、サポートされている文字セットのサブディレクトリがあります。たとえば、*cp850* は *us_english* で使用可能な文字セットです。Sybase 製品のソフトウェア・メッセージ・ファイルは、文字セット・サブディレクトリにあります。

図 *9-4:* メッセージ・ディレクトリ構造



メッセージ言語とグローバル変数

次のグローバル変数には、言語に関する情報が格納されています。

@@langid	現在使用している言語のローカル言語 ID (syslanguages.langid で指定されている) が格納される。
@@language	現在使用している言語の名前 (syslanguages.name で指定されている) が格納される。

第 10 章

クライアント／サーバの文字セット変換の設定

文字セット変換

異機種間環境では、Adaptive Server は、異なるプラットフォームで稼働し、異なる文字セットを使用しているクライアントと通信することがあります。異なる文字セットが同じ言語をサポートしていることがありますが (たとえば、ISO 8858-1 と CP 850 はグループ 1 の言語をサポートしています)、それらの文字セットは同じ文字を異なる方法でコード化する場合があります。たとえば、ISO 8859-1 では、*à* という文字は 16 進の *0xE0* にコード化されます。しかし、CP 850 では、この文字は 16 進数の *0x85* にコード化されます。

クライアント／サーバ間のデータの整合性を維持するには、データを文字セット間で変換する必要があります。これは、マシンや文字セットの種類が変わっても、“a” が常に “a” となるようにするためです。このプロセスを「文字セット変換」と呼びます。

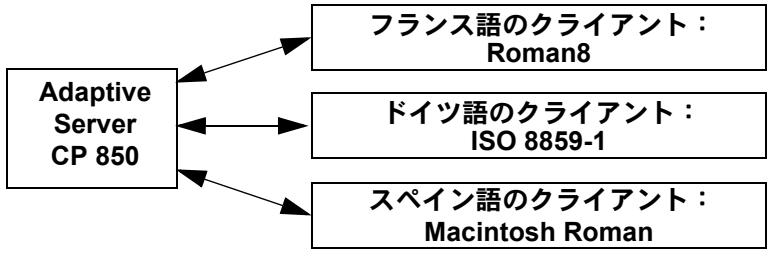
サポートする文字セット変換

文字セット変換は、1 対の文字セット間で行われます。それぞれのクライアント／サーバ・システムでサポートする変換は、そのサーバとクライアントで使用する文字セットにより異なります。サーバがネイティブな文字セットをデフォルトとして使用する場合と、Unicode UTF-8 を使用する場合とでは、異なるタイプの文字セット変換が行われます。

ネイティブな文字セットでの変換

Adaptive Server では、同じ言語グループに属するネイティブな文字セット間での変換がサポートされています。あるネイティブな文字セットがサーバのデフォルトとして設定されている場合、クライアントの文字セットも同じ言語グループに属している必要があります。図 10-1 は、西欧のサーバ／クライアント・システムの例を示しています。この例では、クライアントの文字セットと Adaptive Server のデフォルトの文字セットはすべて言語グループ 1 に属しています。クライアントの文字セットとサーバのデフォルトの文字セットの間で正しくデータが変換されています。クライアントはすべて同じ言語グループに属しているため、どのクライアントから送信されたデータであっても、サーバ上のデータはすべてのクライアントで表示できます。

図 *10-1:* サーバとクライアントの文字セットが同じ言語グループに属している場合の文字セット変換

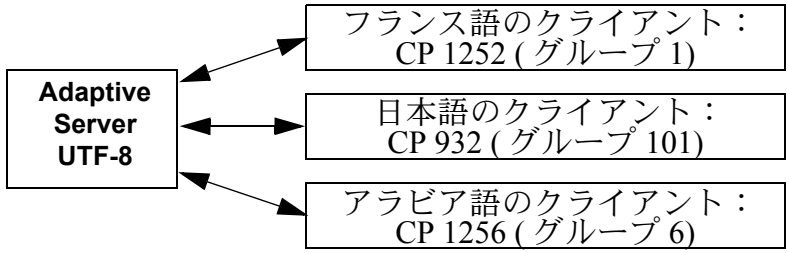


言語グループとサポートされる文字セットの一覧については、表 9-1 (288 ページ) を参照してください。

Unicode システムでの変換

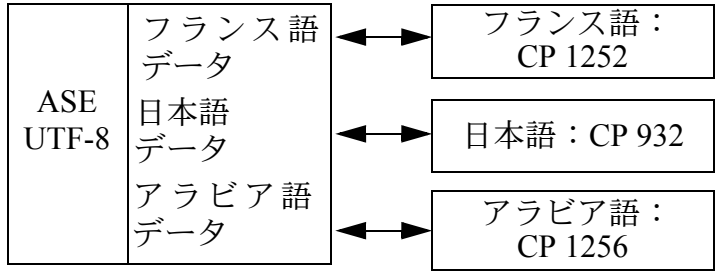
Adaptive Server では、Sybase がサポートするすべてのネイティブな文字セットと UTF-8 との間の文字セット変換もサポートされています。Unicode システムでは、サーバのデフォルトの文字セットが UTF-8 であるため、クライアントの文字セットは、どの言語グループのネイティブ文字セットであってもかまいません。したがって、日本語のクライアント (グループ 101)、フランス語のクライアント (グループ 1)、アラビア語のクライアント (グループ 6) はいずれも、同じサーバとの間でデータの送受信が可能です。それぞれのクライアントからのデータは、各クライアントとサーバ間での受け渡し時に、正しく変換されます。

図 *10-2: Unicode* システムでの文字セット変換



それぞれのクライアントで表示できるのは、その文字セットでサポートされている言語のデータだけです。したがって、日本語のクライアントは、サーバ上の日本語のデータはすべて表示できますが、アラビア語やフランス語のデータは表示できません。同様に、フランス語のクライアントは、フランス語の他に、文字セットでサポートされている西欧言語を表示できますが、日本語やアラビア語は表示できません。

図 *10-3: Unicode* データの表示



追加文字セットである ASCII-7 は、Unicode をはじめ すべての文字セットのサブセットであり、したがって、すべての言語グループのすべての文字セットと互換性があります。Adaptive Server またはクライアントのどちらかの文字セットが ASCII-7 ならば、7 ビットの ASCII 文字をクライアントとサーバ間で送受信するときに変更も変換も必要ありません。

サーバの文字セットを ASCII-7 に設定することはおすすめしません。ただし、各クライアントに対して、それぞれのネイティブな文字セットの最初の 128 文字だけを使用するように制限すると、同じ互換性が得られます。

Adaptive Server 直接変換

Adaptive Server 直接変換は、同じ言語グループの 2 つのネイティブな文字セット間での変換です。たとえば、CP 437 と CP 850 は両方ともグループ 1 の言語グループに属しているため、Adaptive Server はこれらの間での変換をサポートします。Adaptive Server 直接変換は、同じ言語グループの数多くの (すべてではない) ネイティブの文字セット間で行われます (表 10-1 (327 ページ) 参照)。

Unicode 変換

Unicode 変換は、すべてのネイティブな文字セットについて行われます。2 つのネイティブな文字セット間の Unicode 変換では、Unicode が中間文字セットとして使用されます。たとえば、サーバのデフォルトの文字セット (CP 437) とクライアントの文字セット (CP 860) との間で変換する場合、最初に CP 437 が Unicode に変換され、次に Unicode が CP 860 に変換されます。

Unicode 変換は、サーバのデフォルトの文字セットが UTF-8 またはネイティブな文字セットである場合に使用されます。Unicode 変換を使用するには、サーバ設定でそのことを明示的に指定してください (サーバのデフォルトの文字セットが UTF-8 でない場合)。

以前のバージョンの Adaptive Server では直接変換が使用されており、現在でも文字セット変換のデフォルトの方法となっています。ただし、Unicode 変換では、文字セット変換が簡単になり、複雑でなくなります。Adaptive Server 直接変換は引き続きサポートされていますが、Sybase では、すべての文字セットの変換を完全にサポートできるようにするために Unicode 変換も使用しており、新しい直接変換を追加する予定はありません。

変換タイプの選択

実際のクライアント／サーバ・システムでどちらの変換が利用可能かを判断するには、表 10-1 (327 ページ) を参照してください。

非 Unicode クライアント／サーバ・システム

非 Unicode システムでは、サーバ／クライアントの文字セットはネイティブな文字セットなので、Adaptive Server 直接変換を使用できます。

しかし、いくつかの文字セットについては、Adaptive Server 直接変換が用意されていないことがあります。その場合は、Unicode 変換を使用してください。

- クライアント／サーバ・システムのすべての文字セットが、表 10-1 のカラム 1 にある場合は、Adaptive Server 直接変換を使用してください。すべての文字セットが、同じ言語グループに属している必要があります。
- クライアント／サーバ・システムの文字セットが、すべて表 10-1 のカラム 2 にある場合、またはカラム 1 と 2 の組み合わせである場合は、Unicode 変換を使用するようにサーバを設定してください。この場合も、すべての文字セットが同じ言語グループに属している必要があります。

たとえば、サーバのデフォルトの文字セットが CP 850 で、クライアントの文字セットが ISO 8859-1 と ROMAN 8 のいずれかであるとします。表 10-1 を参照すると、CP 850 とクライアントの文字セットの間では直接変換が行われることがわかります。次に、CP 1252 を使用するクライアントを追加したとします。CP 1252 と CP 850 (サーバのデフォルト文字セット) の間には直接変換がないので、CP 1252 と CP 850 の間の変換には Unicode 変換を使用する必要があります。文字セットが混在している (ある文字セットは Adaptive Server 直接変換を使用でき、ある文字セットは Unicode 変換を使用する必要がある) ときは、Adaptive Server 直接変換と Unicode 変換を組み合わせて使用することを指定できます。

Unicode クライアント／サーバ・システム

サーバのデフォルトの文字セットとして Unicode UTF-8 を使用している場合は、変換はすべて UTF-8 と、クライアント・システムで使用されているネイティブ文字セットとの間で行われます。Unicode システムでは、Unicode 変換だけが使用されます。

表 *10-1:* 文字セットの変換方法

言語グループ	カラム 1 － Adaptive Server 直接変換と Unicode 変換	カラム 2 － Unicode 変換のみ
グループ 1	CP 437、CP 850、ISO 8859-1、Macintosh Roman	CP 860、CP 1252、ISO 8859-15、CP 863
グループ 2	CP 852、CP 1250、CP 8859-1、Macintosh Central European	ISO 8859-2
グループ 4	変換不要 (1 つの文字セットだけをサポート)	
グループ 5	CP 855、CP 866、CP 1251、ISO 8859-5、Koi8、Macintosh Cyrillic	
グループ 6		CP 864、CP 1256、ISO 8859-6
グループ 7	CP 869、CP 1253、GREEK8、ISO 8859-7、Macintosh Greek	
グループ 8		CP 1255、ISO 8859-8
グループ 9	CP 857、CP 1254、ISO 8859-9、Macintosh Turkish、TURKISH8	
グループ 101	DEC Kanji、EUC-JIS、Shift-JIS	CP 932
グループ 102		CP 936、EUG-GB、GB18303
グループ 103		Big 5、CP 950、EUC-CNS
グループ 104		EUCKSC、CP 949
グループ 105		CP 874、TIS 620
グループ 106	変換不要 (1 つの文字セットだけをサポート)	
Unicode	変換不要 (1 つの文字セットだけをサポート)	

サーバの設定

デフォルトでは、Adaptive Server は直接変換を使用して異なる文字セット間でデータを変換します。Unicode 変換を使用するには、sp_configure を使用して enable unicode conversions オプションを 1 または 2 に設定します。

- sp_configure “enable unicode conversions” を 1 に設定した場合

  この設定では、Adaptive Server 直接変換または Unicode 変換を使用します。Adaptive Server は最初に、サーバの文字セットとクライアントの文字セットの間で使用できる Adaptive Server 直接変換があるかどうかをチェックします。直接変換がある場合は直接変換を使用し、ない場合は Unicode 変換を使用します。

  クライアント／サーバ・システムの文字セットが、表 10-1 のカラム 1 と 2 の両方に分かれている場合に、この設定を使用してください。

- sp_configure “enable unicode conversions” を 2 に設定した場合

  この設定では、Unicode 変換だけを使用します。Adaptive Server は、使用可能な Adaptive Server 直接変換があるかどうかをチェックしないで、Unicode 変換を使用します。

  クライアント／サーバの変換によってデータ長の変更が生じる場合に、この設定を使用してください (「変換とデータ長の変更」(330 ページ) 参照)。

すべての文字セットが表 10-1 のカラム 2 に該当する場合は、enable unicode conversions を 2 に設定して、常に Unicode 変換が使用されるようにします。

Adaptive Server バージョン 15.0 以降では、enable unicode conversions のデフォルト値は 1 です。

サーバのデフォルトの文字セットが UTF-8 である場合は、サーバは自動的に Unicode 変換だけを使用します。

文字セット変換の有効化と無効化

接続を要求するクライアントは、その文字セットを Adaptive Server に知らせます。Adaptive Server はクライアントの文字セットをデフォルトの文字セットと比較し、2 つの名前が同一であれば、変換を要求しません。名前が異なる場合は、Adaptive Server のデフォルトの文字セットとクライアントの文字セットの間の変換をサポートしているかどうかを判断します。サポートしていない場合は、エラー・メッセージをクライアントに送り、ログイン処理を続行します。サポートしている場合は、文字セット変換が自動的に有効になります。サーバのデフォルトの文字セットが UTF-8 である場合は、Unicode 変換が自動的に使用されます。デフォルトがネイティブな文字セットの場合、ユーザが Unicode 変換を要求していないかぎり、サーバは Adaptive Server 直接変換を使用します。

文字セット変換は、サーバ・レベルで無効にできます。以下のような場合に無効にします。

- すべてのクライアントがサーバのデフォルトと同じ文字セットを使用しているため、変換する必要がない場合。
- クライアントの文字セットとサーバのデフォルトの文字セットとの変換がサポートされていない場合。
- コード化を変更しないで、データをサーバに保存する場合。

サーバ・レベルで文字セット変換を無効にするには、disable character set conversion パラメータを 1 に設定します。

また、クライアント・セッション内で set char_convert コマンドを使用することによって、文字セット変換を接続レベルでも制御できます。set char_convert off は、特定のクライアントとサーバの間の変換を無効にします。クライアントとサーバが同じ文字セットを使用している場合は変換が不要になるため、set char_convert off を使用してください。set char_convert on を実行すると、変換は再び有効になります。

変換できない文字

次の場合に、一部の文字が変換されないことがあります。

- 文字が、変換元の文字セットには存在している (コード化されている) が、ターゲットの文字セットには存在しない場合。たとえば、Macintosh の文字セット中にある OE の合字 (コード・ポイント 0xCE) です。この文字は、ISO 8859-1 文字セットには存在しません。Macintosh から ISO 8859-1 文字セットに変換するデータに OE の合字が存在すると、変換エラーが起こります。
- 文字が、変換元の文字セットにもターゲットの文字セットにも存在するが、その文字を表すためのバイト数が変換元の文字セットとターゲットの文字セットとで異なる場合。

  たとえば、1 バイトのアクセント記号付き文字 (á や è など) は、UTF-8 では 2 バイト文字であり、2 バイトの**タイ語**の文字は UTF-8 では 3 バイト文字になります。この制限を回避するには、enable unicode conversion オプションを 1 または 2 に設定します

文字セット変換のエラー処理

ある文字がクライアントの文字セットに存在してサーバの文字セットには存在しない、またはその逆の場合は、Adaptive Server の文字セット変換の変換エラーがレポートされます。Adaptive Server は、サーバへの入力時に正常に変換されたデータをクライアントが取り出すときに、クライアントの文字セットに正常に再変換できることを保証しなければなりません。そのためには、データベースに疑わしいデータが保管されることがないようにする必要があります。

データの入力中に変換エラーが発生すると、次のメッセージが生成されます。

```
Msg 2402, Severity 16 (EX_USER):
Error converting client characters into server's
character set. Some character(s) could not be converted.
```

変換エラーが発生すると、挿入文や更新文が含まれているクエリは実行できません。エラーが発生した場合は、データを見直して問題の文字を探し、置換してください。

Adaptive Server からクライアントにデータを送信するときに変換エラーが検出された場合は、疑わしい文字のバイトは ASCII の疑問符 (?) に置き換えられます。クエリ・バッチは最後まで実行されます。文が完了すると、Adaptive Server は次のメッセージを送信します。

```
Msg 2403, Severity 16 (EX_INFO):
WARNING! Some character(s) could not be converted into client's
character set. Unconverted bytes were changed to question marks
('?').
```

変換とデータ長の変更

場合によっては、サーバの文字セットとクライアントの文字セットの間でデータを変換した結果、データ長に変更が生じることがあります。たとえば、一方のシステムの文字セットでは 1 つの文字を表すために 1 バイトを使用し、もう一方のシステムの文字セットでは 1 つの文字につき 2 バイトが必要になる場合に変更が生じます。

文字セット変換によってデータ長に変更が生じた場合、次の 2 つの状態が考えられます。

- 次の例のように、データ長が減少する。
  - ギリシャ語またはロシア語での、マルチバイトの UTF-8 からシングルバイトのギリシャ語またはロシア語文字セットへの変換
  - 日本語での、EUC-JIS の 2 バイト半角カタカナ文字から Shift-JIS のシングルバイトの文字への変換

- 次の例のように、データ長が増加する。
  - シングルバイトのタイ語文字から UTF-8 でのマルチバイトのタイ語文字への変換
  - Shift-JIS での日本語のシングルバイト文字から EUC-JIS での 2 バイト半角カタカナ文字への変換

システムとアプリケーションの設定

クライアント／サーバ・システムの中に UTF-8 を使用している部分がある場合や、日本語文字セットを使用している場合は、文字セット変換の結果としてデータ長の変更が生じる可能性があります。データ長の変更を処理できるようにサーバを設定してください。また、場合によっては、クライアントでもデータ長の変更を処理できるように設定する必要があります。

1 Unicode 変換を使用するようにサーバを設定します。「サーバの設定」(328 ページ) を参照してください。サーバとクライアントの間でデータ長が増加する場合は、手順 2 と 3 も行ってください。

2 クライアントは、11.1 以降の Open Client を使用している必要があります。クライアントが CS_LONGCHAR データを処理できることを、Open Client の ct_capability を使用して、接続時にサーバに知らせる必要があります。

capability パラメータは CS_DATA_LCHAR に設定し、*value* パラメータは CS_TRUE に設定してください。*connection* は、CS_CONNECTION 構造体へのポインタです。

```
CS_INT capval = CS_TRUE
ct_capability(connection,CS_SET,CS_CAP_RESPONS,
     CS_DATA_LCHAR,&capval)
```

3 変換によってデータ長が増加した場合、char および varchar のデータは、クライアントの文字セットに変換され、CS_LONGCHAR データとしてクライアントに送信されます。CS_LONGCHAR として受信したデータを抽出するようにクライアント・アプリケーションをコーディングする必要があります。

ユーティリティ・プログラムのための文字セットの指定

Sybase のユーティリティ・プログラムは、クライアント・プラットフォームのデフォルトの文字セットが、クライアントの使用している文字セットと同一だと見なします。ただし、クライアントの文字セットが、プラットフォームの文字セットと異なる場合もあります。このため、コマンド・ラインでクライアントの文字セットを指定しなければならないことがあります。isql、bcp、defncopy の各ユーティリティのコマンド・ラインのオプションとしてクライアントの文字セットを指定し、LANG 変数の設定や *locales.dat* の設定を一時的に無効にします。

-J *charset_name* (UNIX と PC の場合) を指定すると、クライアントの文字セットは *charset_name* であるとして設定されます。

クライアントの文字セットのコマンド・ライン・フラグを省略すると、プラットフォームのデフォルトの文字セットが使用されます。『ASE ユーティリティ・ガイド』を参照してください。

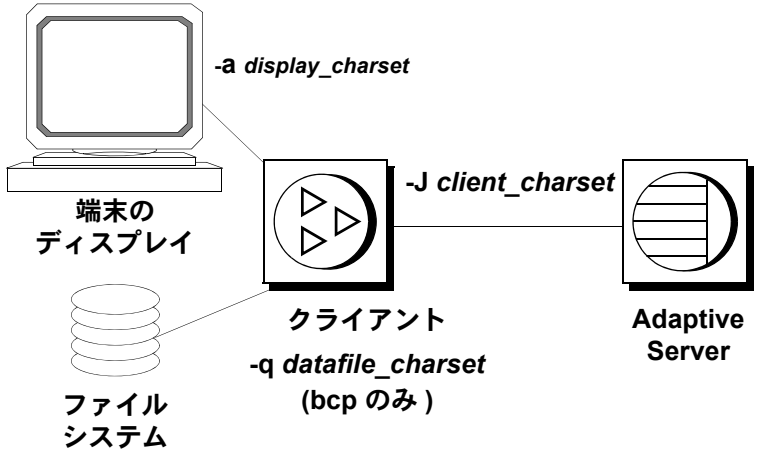
表示およびファイル文字セットのコマンド・ライン・オプション

この章では、クライアントと Adaptive Server の間の文字セットの変換について説明していますが、次の 2 つについても文字セットの変換が必要になることがあります。

- クライアントと端末間
- クライアントとファイル・システム間

図 10-4 は、スタンドアロン・ユーティリティ isql、bcp、defncopy で利用できるパスとコマンド・ライン・オプションを示しています。

図 *10-4:* 文字セットの変換が必要な場所



-J または /clientcharset コマンド・ライン・オプションによって、クライアントが Adaptive Server との文字データの送受信に使用する文字セットを指定します。

表示文字セットの設定

クライアントを実行する端末の文字セットがクライアントの文字セットとは異なる場合は、-a コマンド・ライン・オプションを使用してください。図 10-4 では、変換に必要な文字セット変換ファイル (*.xlt* ファイル) を指定するために、-a オプションと -J オプションを一緒に使用しています。

-a を使用するときに -J を省略できるのは、クライアントの文字セットがデフォルトの文字セットと同じ場合だけです。

ファイル文字セットの設定

クライアントの文字セットとは異なる文字セットを使用するファイル・システムとの間で文字データをコピーするために bcp を実行する場合は、-q コマンド・ライン・オプションを使用してください。図 10-4 では、変換に必要な文字セット変換ファイル (*.xlt* ファイル) を指定するために、-q オプションまたは /filecharset オプションと、-J オプションまたは /clientcharset オプションを一緒に使用しています。

第 11 章

システムの問題の診断

この章では、システムの問題の診断および解決方法について説明します。

Adaptive Server のエラー・メッセージ

Adaptive Server は、問題を検出すると、次の内容を含むエラー・メッセージを表示します。

- エラー・メッセージをユニークに識別する「メッセージ番号」
- 問題の種類と重大度レベルを示す 10 ～ 24 の範囲の「重大度レベル番号」
- 「エラー・ステータス番号」。エラーが発生した Adaptive Server コードの行を特定できます。
- 「エラー・メッセージ」。発生した問題の内容を通知します。その問題の解決方法が提示されることもあります。

たとえば、存在しないテーブルにアクセスしようとすると、次のような情報が表示されます。

```
select * from publisher

Msg 208, Level 16, State 1:
publisher not found.Specify owner.objectname or use
sp_help to check whether the object exists (sp_help may
produce lots of output).
```

1 つのクエリに対して複数のエラー・メッセージが発行される場合もあります。バッチまたはクエリで複数のエラーが発生すると、Adaptive Server は通常は最初のエラーだけをレポートします。あとのエラーは、次回のバッチまたはクエリの実行時にレポートされます。

エラー・メッセージは master..sysmessages に保管されており、Adaptive Server のバージョンが新しくなるたびに更新されます。次は、その最初の部分です (Adaptive Server のデフォルト言語が us_english の場合)。

```
select error, severity, description
from sysmessages
where error >=101 and error <=106
and langid is null
```

```
error severity description
----- -------- ----------------------------------------------
  101       15 Line %d: SQL syntax error.
  102       15 Incorrect syntax near '%.*s'.
  103       15 The %S_MSG that starts with '%.*s' is too long.
               Maximum length is %d.
  104       15 Order-by items must appear in the select-list if
               the statement contains set operators.
  105       15 Unclosed quote before the character string '%.*s'.
  106       16 Too many table names in the query. The maximum
               allowable is %d.

(6 rows affected)
```

sysmessages を問い合わせて、エラー・メッセージのカスタム・リストを生成できます。

- サーバで複数の言語がサポートされている場合、sysmessages には言語ごとに各メッセージが保管されています。langid カラムは、us_english では NULL で、サーバにインストールされている他の言語の場合は syslanguages.langid に一致します。
- sysmessages の dlevel カラムは、現在使用されていません。
- sqlstate カラムは、ANSI SQL92 で定義されている、エラー条件と例外の SQLSTATE 値を保管します。
- 17000 以上のメッセージ番号は、システム・プロシージャのエラー・メッセージとメッセージ文字列です。

エラー・メッセージおよびメッセージ番号

メッセージ番号 (error) と言語 ID (langid) の組み合わせによって、各エラー・メッセージがユニークに識別されます。同じメッセージ番号で異なる言語 ID のメッセージは、翻訳されたメッセージであることを示します。

```
select error, description, langid
from sysmessages
where error = 101

error description                                langid
----- ---------------------------------------- ------
  101 Line %d: SQL syntax error.                 NULL
  101 Ligne %1!: erreur de syntaxe SQL.             1
  101 Zeile %1!: SQL Syntaxfehler.                  2

(3 rows affected)
```

エラー・メッセージのテキストは、問題の説明です。この説明には、行番号、データベース・オブジェクトのタイプ (テーブル、カラム、ストアド・プロシージャなど) の参照、特定のデータベース・オブジェクト名などが含まれることもあります。

sysmessages の description フィールドには、パーセント記号 (%) に 1 文字以上の文字列が続くことがありますが、これはプレースホルダの役割を果たすもので、Adaptive Server が問題を検出してそのエラー・メッセージを生成するときにデータが代入されます。“%d” は数値のプレースホルダです。“%S_MSG” はデータベース・オブジェクトの種類を表すプレースホルダです。引用符で囲まれた “%.*s” は、特定のデータベース・オブジェクト名のプレースホルダです。表 11-1 (338 ページ) に、プレースホルダとその意味を示します。

たとえば、メッセージ番号 103 の description フィールドは次のとおりです。

```
The %S_MSG that starts with '%.*s' is too long. Maximum length
is %d.
```

実際に表示されるエラー・メッセージは次のようになります。

```
The column that starts with 'title' is too long. Maximum length
is 80.
```

Sybase 製品の保守契約を結んでいるサポート・センタにエラーをレポートするときは、番号、オブジェクト・タイプ、およびオブジェクト名もレポートしてください (「エラーのレポート」(346 ページ) を参照)。

エラー・メッセージ・テキスト内の変数

表 11-1 は、エラー・メッセージ・テキストに使用される記号とその意味を示しています。

表 *11-1:* エラー・テキストに使用される記号

記号	意味
%d、%D	10 進数
%x、%X、%.*x、%lx、%04x、%08lx	16 進数
%s	null で終了する文字列
%.*s、%*s、%*.s	文字列 (通常は特定のデータベース・オブジェクト名)
%S_*type*	Adaptive Server で定義されている構造体
%c	1 つの文字
%f	浮動小数点数
%ld	長い 10 進数
%lf	倍精度浮動小数点数

Adaptive Server エラー・ロギング

Adaptive Server から出力されるエラー・メッセージは、ユーザの画面だけに送信されます。

致命的なエラー・メッセージ (重大度レベル 19 以上) からのスタック・トレース、およびカーネルからのエラー・メッセージは、エラー・ログ・ファイルにも送信されます。このファイルの名前については、プラットフォームの『Adaptive Server Enterprise 設定ガイド』または『ASE ユーティリティ・ガイド』を参照してください。

注意 エラー・ログ・ファイルの所有者は、Adaptive Server をインストールしたユーザ (またはエラー・ログが削除された後に Adaptive Server を起動したユーザ) です。オペレーティング・システム・レベルでのエラー・ログのパーミッションまたは所有権に問題があると、Adaptive Server が正常に起動できないことがあります。

エラー・ログが存在しない場合は、Adapitve Server によってエラー・ログが作成されます。エラー・ログのロケーションは、起動時に runserver ファイル内の *errorlogfile* パラメータとして、またはコマンド・ラインで指定します。別のロケーションが選択された場合を除いて、Sybase インストーラ・ユーティリティは、*$SYBASE/install* をエラー・ログのロケーションとして runserver ファイルを設定します。runserver ファイルまたはコマンド・ラインでロケーションを指定しなければ、エラー・ログは Adaptive Server を起動したディレクトリに作成されます。エラー・ログのロケーションの指定については、『ASE ユーティリティ・ガイド』の dataserver を参照してください。

注意　エラー・ログをすぐに参照できるように、Adaptive Server を常に同じディレクトリから起動するか、runserver ファイルまたはエラー・ログ・フラグを使用してください。

サーバを起動したときに、サーバ上の各データベースの起動およびリカバリが正常に行われたか失敗したかは、エラー・ログ内のメッセージによってわかります。エラー・ログ・ファイルには、以降の致命的エラーのメッセージとすべてのカーネル・エラー・メッセージが追加されていきます。古いメッセージや不要なメッセージを削除してエラー・ログのサイズを小さくするには、Adaptive Server の停止中に「削除」を行ってください。

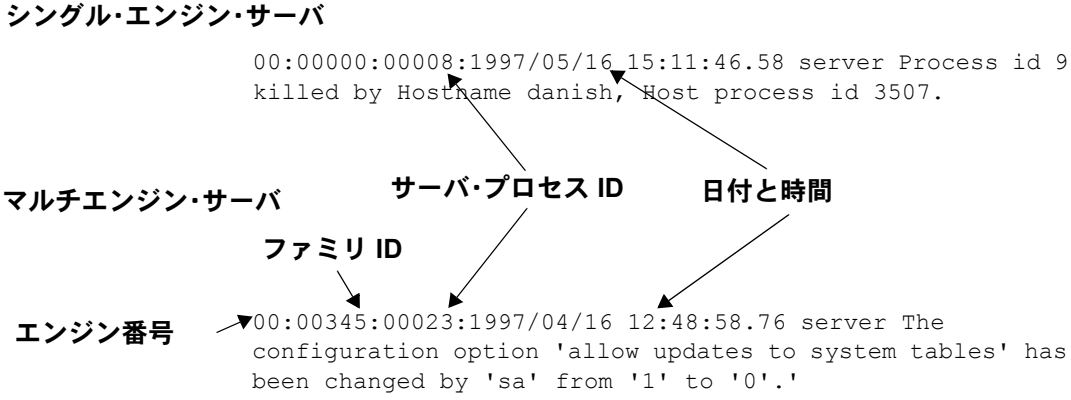
エラー・ログのフォーマット

エラー・ログのエントリには、次の情報が含まれています。

- 各ログ・エントリに関連したエンジン。エンジン番号は、2 桁の数で表されます。オンラインのエンジンが 1 つだけの場合は、“00” が表示されます。
- (発生した) スレッドのファミリ ID
  - 逐次処理では、“00000” が表示される。
  - 並列処理では、(発生した) スレッドの親の SPID 番号が表示される。
- (発生した) スレッドの SPID
  - 逐次処理では、メッセージを生成したスレッドの SPID 番号が表示される。スレッドがシステム・タスクの場合は、“00000” が表示される。
  - 並列処理では、(発生した) スレッドの SPID 番号が表示される。
- `yyyy/mm/dd` の形式の日付。これを利用すると、エラー・メッセージを日付順にソートできます。
- 24 時間形式で表された時刻。時刻には、秒および 1/100 秒までが含まれます。
- “server” または “kernel”。これは、Sybase 製品の保守契約を結んでいるサポート・センタ専用のエントリです。
- エラー・メッセージ自体

図 11-1 は、エラー・ログの例を示しています。

図 *11-1:* エラー・ログのフォーマット



重大度レベル

メッセージの重大度レベルは、Adaptive Server が検出した問題の種類と重大度を示すものです。整合性を保つため、Adaptive Server はエラー状態が発生したときに sysmessages からメッセージを表示しますが、処理は内部テーブルに従って行います。エラーに対応するメッセージの重大度がそれぞれ異なることがあるため、開発するアプリケーションまたはプロシージャで Adaptive Server のメッセージや重大度レベルを参照する場合に、想定した動作が異なっている可能性があります。

警告！ Adaptive Server のエラー番号に基づいて、独自のエラー番号とメッセージを作成できます (たとえば、Adaptive Server の値に 20,000 を追加します)。ただし、sysmessages システム・テーブルにある、Adaptive Server によって提供されるシステム・メッセージを変更することはできません。

ユーザ定義のエラー・メッセージを sysusermessages に追加するには、sp_addmessage を使用します。『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。

重大度レベルが 17 以上の問題が発生した場合は、ユーザは必ずシステム管理者に連絡します。システム管理者には、そのエラーを解決し、その発生頻度を監視する責任があります。

その問題の影響がデータベース全体に及ぶ場合、システム管理者は、データベース一貫性チェッカ (dbcc) を使用して損傷の範囲を判断しなければならない場合があります。dbcc によって、削除すべきオブジェクトを識別できることがあります。dbcc によって損傷を修復できることもありますが、データベースを再ロードする必要が生じることもあります。

詳細については、『システム管理ガイド 第 2 巻』の次の章を参照してください。

- dbcc については、『システム管理ガイド 第 2 巻』の「第 10 章 データベースの一貫性の検査」を参照してください。
- ユーザ・データベースのロードについては、『システム管理ガイド 第 2 巻』の「第 12 章 ユーザ・データベースのバックアップとリストア」を参照してください。
- システム・データベースのロードについては、『システム管理ガイド 第 2 巻』の「第 13 章 システム・データベースのリストア」を参照してください。

重大度レベル 10 ～ 18

重大度レベルが 10 ～ 16 のエラー・メッセージは、問題の原因がユーザ・エラーである場合に生成されます。これらは、ユーザが解決できるエラーです。重大度レベル 17 と 18 のエラーでは、ユーザのセッションは停止されません。

重大度レベルが 17 以上のエラー・メッセージが発生したときは、システム管理者またはデータベース所有者へのレポートが必要です。

重要度レベル 10：ステータス情報

重大度レベルが 10 のメッセージは、エラーではありません。特定のコマンドの実行後に追加情報を表示するためのもので、通常はメッセージ番号や重大度レベルは表示されません。たとえば create database コマンドを実行すると、要求した領域のどの程度が新しいデータベースに割り付けられたかを示すメッセージが表示されます。

重大度レベル 11：指定されたデータベース・オブジェクトが見つからない

重大度レベル 11 のメッセージは、コマンドで指定されたオブジェクトを Adaptive Server が見つけることができないことを示します。

主な原因には、データベース・オブジェクト名の入力ミス、オブジェクトの所有者名の指定もれ、現在のデータベースの誤認などがあります。オブジェクト名が正しく入力されていることを確認し、オブジェクトの所有者がユーザ自身または “dbo” でない場合は所有者名を指定します。また、現在のデータベースが正しいことを確認してください。

重大度レベル 12：不正データ型の検出

重大度レベル 12 のメッセージは、データ型に問題があることを示します。たとえば、カラムに正しくないデータ型の値を入力しようとした場合や、比較するカラムどうしのデータ型が異なり、互換性もない場合です。

比較の問題を解決するには、select 文で convert 関数を使用してください。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』または『Transact-SQL ユーザーズ・ガイド』を参照してください。

重大度レベル 13：ユーザ・トランザクションの構文エラー

重大度レベル 13 のメッセージは、現在のユーザ定義トランザクションに問題があることを示します。たとえば、begin transaction を発行しないで commit transaction コマンドを発行した場合や、定義されていないセーブポイントにトランザクションをロールバックしようとした場合です (セーブポイント名の入力ミスの場合もあります)。

重大度レベル 13 はデッドロックを示すこともあります。このとき、デッドロック・ビクティムとなったプロセスはロールバックされます。ユーザは、コマンドを初めから実行し直す必要があります。

重大度レベル 14：コマンド実行のパーミッションが不十分

重大度レベル 14 のメッセージは、コマンドの実行やデータベース・オブジェクトへのアクセスに必要なパーミッションがユーザにないことを示します。データベース・オブジェクトの所有者、データベースの所有者、またはシステム管理者に連絡して、問題のコマンドやオブジェクトを使用するためのパーミッションを取得してください。

重大度レベル 15：SQL 文の構文エラー

重大度レベル 15 のメッセージは、コマンドの構文に誤りがあることを示します。このエラー・メッセージのテキストには、誤りのある行番号およびその付近のキーワードが含まれています。

重大度レベル 16：その他のさまざまなユーザ・エラー

重大度レベル 16 のエラー・メッセージのほとんどは、他のカテゴリに属さない、致命的でない誤りがあったことを示します。重大度レベルが 16 以上の場合は、ソフトウェアまたはハードウェアのエラーを示すこともあります。

たとえば、制約に違反する方法でビューを更新しようとした場合です。また、コマンドの中でカラム名を修飾せずに使用したとき、同じ名前のカラムがそのコマンドで使用する別のテーブルにもある場合に、このカテゴリのエラーとなります。Adaptive Server は、ユーザがどのテーブルを使用したいのかは判断できません。コマンドの構文と作業データベース・コンテキストを確認してください。

通常は重大度レベルが 17 以上となるメッセージでも、dbcc checktable または dbcc checkalloc によって発生した場合は、次のオブジェクトのチェックに進むことができるように重大度レベルは 16 となります。dbcc ユーティリティの実行時に重大度レベル 16 の 2500 ～ 2599 のエラー・メッセージが表示された場合は、『ASE トラブルシューティング＆エラー・メッセージ・ガイド』を参照してください。

注意　重大度レベル 17 および 18 は、通常はエラー・ログにレポートされません。ユーザには、重大度レベル 17 および 18 のエラーが発生した場合はシステム管理者に連絡するように指示してください。

重大度レベル 17：リソース不足

重大度レベル 17 のエラー・メッセージは、コマンドの実行によって Adaptive Server のリソース不足が発生したか、システム管理者が設定した制限を超えたことを示します。ユーザは作業を続行できますが、実行できないコマンドもあります。

システムの制限には、同時にオープンできるデータベースの数や、Adaptive Server への接続数などがあります。制限はシステム・テーブルに保管され、sp_configure コマンドを使用してチェックできます。設定パラメータの変更の詳細については、「第 5 章 設定パラメータ」を参照してください。

重大度レベル 17 のエラー・メッセージが示すエラーが領域不足である場合は、データベース所有者がこの問題を解決できます。その他の重大度レベル 17 のエラー・メッセージについては、解決するのはシステム管理者です。

重大度レベル 18：致命的でない内部エラーが検出された

重大度レベル 18 のエラー・メッセージは、内部ソフトウェアのバグを示します。ただし、コマンドは最後まで実行され、Adaptive Server との接続は維持されます。ユーザは実行中の作業を続行できますが、実行できないコマンドもあります。重大度レベル 18 のエラーが発生する状況には、クエリのアクセス・パス決定の理由が正当なものではないことを Adaptive Server が検出した場合があります。

このようなメッセージを表示する問題が発生しても、ユーザの作業は中断されないので、その問題がユーザからレポートされないことがあります。ただし、システム管理者が問題をレポートできるようにするため、重大度レベル 18 以上のエラー・メッセージが表示された場合は必ずシステム管理者に報告するように、ユーザに指示を与えてください。

重大度レベル 19 ～ 26

致命的な問題が発生すると、重大度レベル 19 以上のエラー・メッセージが生成されます。このとき、Adaptive Server とユーザとの接続は切断されます (重大度レベルが高い場合は Adaptive Server が停止することがあります)。作業を続けるには、ユーザはクライアント・プログラムを再起動しなければなりません。

致命的なエラーが発生すると、プロセスは、停止する前にいったん静止状態となり、発生した問題についての情報を記録します。その後、プロセスは強制終了されて消滅します。

ユーザの接続が切断されたとき、ユーザは再接続して作業を再開できないこともあります。この範囲の重大度レベルの問題の影響は、1 ユーザの 1 プロセスにとどまらず、データベース内のプロセス全体に及ぶこともあり得ます。場合によっては、システム管理者が Adaptive Server を再起動する必要があります。これらの問題は、必ずと言うわけではありませんが、データベースやそのオブジェクトに損傷を与えることがあります。また逆に、データベースやそのオブジェクトの損傷によりこれらの問題が発生する場合もあります。ハードウェアの故障が原因の問題もあります。

カーネルからのエラー・メッセージは、エラー・ログ・ファイルに送信されます。

重大度レベル 19：リソースでの Adaptive Server の致命的なエラー

重大度レベル 19 のエラー・メッセージは、設定可能でない内部制限値を超えたこと、および Adaptive Server が正常にリカバリできないことを示します。ユーザは、Adaptive Server に再接続する必要があります。

重大度レベル 20：現在のプロセスでの Adaptive Server の致命的なエラー

重大度レベル 20 のエラー・メッセージは、コマンドのバグが Adaptive Server によって検出されたことを示します。この問題が影響するのは現在のプロセスだけです。また、データベースが損傷を受けている可能性はほとんどありません。dbcc 診断を実行してください。ユーザは、Adaptive Server に再接続する必要があります。

重大度レベル 21：データベース・プロセスでの Adaptive Server の致命的なエラー

重大度レベル 21 のエラー・メッセージは、現在のデータベースでのすべてのプロセスに影響を与えるバグが Adaptive Server によって検出されたことを示します。ただし、データベース自体が損傷を受けている可能性はほとんどありません。Adaptive Server を再起動して、dbcc 診断を実行してください。ユーザは、Adaptive Server に再接続する必要があります。

重大度レベル 22：Adaptive Server の致命的なエラー：テーブルの整合性の損傷

重大度レベル 22 のエラー・メッセージは、メッセージに示されたテーブルまたはインデックスが、以前にソフトウェアまたはハードウェアの問題によって損傷を受けたことを示します。

まず、Adaptive Server を再起動して dbcc を実行し、データベース内の他のオブジェクトも損傷を受けているかどうかを調べてください。dbcc のレポート内容にかかわらず、ディスク自体には問題がなく、キャッシュ内だけに問題が存在する場合があります。この場合は、Adaptive Server を再起動すれば問題は解消されます。

再起動しても問題が解決できない場合は、ディスクにも問題があります。エラー・メッセージに示されたオブジェクトを削除すると、問題が解決できる場合があります。たとえば、ノンクラスタード・インデックスの中に長さ 0 のローが見つかったというメッセージが表示された場合は、テーブル所有者はそのインデックスを削除して作り直します。

Adaptive Server は、リカバリ中に検出した疑わしいページまたはインデックスをオフラインにします。sp_setsuspect_granularity を使用して、リカバリ中に疑わしいと判断されたものがデータベース全体なのか、または個々のページのみなのかを確認してください。『リファレンス・マニュアル：プロシージャ』の「sp_setsuspect_granularity」を参照してください。

ユーザは、Adaptive Server に再接続する必要があります。

重大度レベル 23：致命的なエラー：データベースの整合性の損傷

重大度レベル 23 のエラー・メッセージは、以前にソフトウェアまたはハードウェアの問題によって発生した損傷が原因で、データベース全体の整合性が失われた可能性があることを示します。Adaptive Server を再起動して、dbcc 診断を実行してください。

データベース全体に問題の可能性があることをエラー・メッセージが示していても、実際にはキャッシュだけの損傷で、ディスク自体には問題がないことがあります。その場合は、startserver を使用して Adaptive Server を再起動すると、問題が解決されます。

重大度レベル 24：ハードウェア・エラーまたはシステム・テーブルの損傷

重大度レベル 24 のエラー・メッセージは、メディア障害または (まれに) sysusages の矛盾を示します。この場合、システム管理者がデータベースを再ロードする必要があります。また、ハードウェアの購入元に連絡する必要がある場合もあります。

重大度レベル 25：Adaptive Server 内部エラー

重大度レベル 25 のエラーは画面には表示されません。Adaptive Server 内部エラーとして処理されます。

重大度レベル 26：ルール・エラー

重大度レベル 26 のエラー・メッセージは、内部的なロックまたは同期の規則に違反していることを示します。Adaptive Server をいったん停止して、再起動する必要があります。

エラーのレポート

Sybase サポート・センタにエラーをレポートするときは、次の情報をレポートしてください。

- メッセージ番号、重大度レベル番号、ステータス番号。
- エラー・メッセージに含まれているすべての数値、データベース・オブジェクトのタイプ、またはデータベース・オブジェクト名。
- メッセージが生成されたときの状況、つまりそのとき実行していたコマンド。エラー・ログのハードコピーを提出すると解決に役立ちます。

Backup Server のエラー・ロギング

Adapitve Server と同じように、Backup Server も、エラー・ログが存在していなければエラー・ログを作成します。エラー・ログのロケーションは、起動時に runserver ファイル内の *error_log_file* パラメータとして、またはコマンド・ラインで指定します。インストール時に別のロケーションが選択された場合を除いて、Sybase インストーラは、*$SYBASE/install* をエラー・ログのロケーションとして runserver ファイルを設定します。runserver ファイルまたはコマンド・ラインでロケーションを指定しなければ、エラー・ログは Backup Server を起動したディレクトリに作成されます。エラー・ログに出力されるメッセージを制限するには、backupserver -V オプション (Windows NT では bcksvr -V) を使用します。詳細については、『ASE ユーティリティ・ガイド』の Backup Server に関する項を参照してください。

Backup Server のエラー・メッセージの形式は次のとおりです。

```
MMM DD YYY:Backup Server:N.N.N.N:Message Text
```

Backup Server のメッセージ番号は、N.N.N.N のようにピリオドで区切られた 4 つの整数で構成されています。N.N.N の形式のメッセージは、Open Server によって送信されるものです。

Backup Server のエラー・メッセージの 4 つのコンポーネントは、*major.minor.severity.state* です。

- *major* コンポーネントは、通常、エラーが発生した Backup Server コードの機能領域を示します。
  - 1 － システム・エラー
  - 2 － Open Server のイベント・エラー
  - 3 － Backup Server のリモート・プロシージャ・コール・エラー
  - 4 － I/O サービス・レイヤ・エラー
  - 5 － ネットワークのデータ転送エラー
  - 6 － ボリューム処理エラー
  - 7 － オプションの解析エラー

  major カテゴリ 1 ～ 6 のエラーの原因は、Backup Server の内部エラーまたはさまざまなシステムの問題が考えられます。major カテゴリ 7 のエラーの原因は、ほとんどがダンプ・コマンドまたはロード・コマンドに指定したオプションの誤りです。
- *minor* 番号は、major カテゴリの中で順番に割り当てられます。
- *severity* は、次のいずれかです。
  - 1 － 情報。ユーザのアクションは不要です。
  - 2、3 － セッションに致命的な影響を与える可能性のある、予期されない状態が発生しました。エラーは、使用状況、環境、または内部論理によって、またはこれらの要因が組み合わされて発生しました。
  - 4 － Backup Server の実行に致命的な影響を与える、予期されない状態が発生しました。Backup Server をただちに終了してください。
- *state* コードは、コード内のエラー・レポートのインスタンスに 1 対 1 で対応しています。Backup Server のエラーについて Sybase の保守契約を結んでいるサポート・センタに連絡する必要がある場合、state コードはエラーの正確な原因を判断するのに役立ちます。

プロセスの強制終了

プロセスとは、Adaptive Server によって実行される実行単位のことです。プロセスには、開始時にユニークなプロセス ID が割り当てられ、この ID 番号を spid といいます。この処理を「初期化」と呼びます。この ID 番号は、各プロセスについてのその他の情報とともに master..sysprocesses に保管されます。並列プロセス環境で実行されるプロセスは、子プロセスを作成します。子プロセスには、それぞれ独自の spid があります。spid を作成して割り当てるプロセスには、Adaptive Server の起動、ログイン・タスク、チェックポイント、ハウスキーピング・タスクなどがあります。starting Adaptive Server, login tasks, checkpoints, the housekeeper tasks, and so onsp_who を実行すると、ほとんどの情報を確認できます。

シングルエンジンのサーバ上で sp_who を実行すると、sp_who プロセスが「実行中」であり、他のすべてのプロセスは「実行可能」またはスリープ状態にあることが表示されます。マルチエンジンのサーバでは、エンジンごとに「実行中」のプロセスが 1 つ存在します。

kill コマンドは、進行中のプロセスを強制終了します。プロセスを強制終了する必要がある状況としては、プロセスが他のユーザの作業を妨害しているときに、そのプロセスの実行責任者に連絡がとれない場合があります。プロセスが保持しているロックによって、データベース・オブジェクトへのアクセスがブロックされている場合や、多数のスリープ中のプロセスによって、使用可能なユーザ接続が占有されている場合もあります。システム管理者は、以下を待っているプロセスを含む、ほとんど実行中または「実行可能な」プロセスを強制終了できます。

- waitfor コマンドなどでのアラーム
- ネットワークの送信または受信
- ロック
- ファミリ内の他のプロセスからの同期メッセージ

終了していないすべてのトランザクションを正常にロールバックして、プロセスが使用しているすべてのシステム・リソースを解放できる場合にかぎり、プロセスの強制終了を実行できます。ファミリの一部であるプロセスでは、子プロセスを強制終了すると、ファミリ内のすべてのプロセスも強制終了されます。しかし、最も簡単な方法は親プロセスを強制終了することです。プロセスのファミリの場合は、子プロセスのステータスが `sync sleep` ならば、kill コマンドがより速く検出されます。

表 11-2 は、sp_who がレポートするステータス値と kill コマンドの効果が反映されるタイミングを示しています。

表 *11-2: sp_who* がレポートするステータスの値

ステータス	意味	kill コマンドの効果
recv sleep	ネットワーク読み込みの待機中。	即時
send sleep	ネットワーク送信の待機中。	即時
alarm sleep	次のようなアラーム待ち。 `waitfor delay "10:00"`	即時
lock sleep	ロック取得の待機中。	即時
sync sleep	ファミリ内の他のプロセスからの同期メッセージ待ち。	即時 (ファミリ内の他のプロセスも強制終了可能な状態にする)
Sleeping	ディスク I/O またはその他のリソース待ち。おそらく、プロセスは実行中であるが非常に大量のディスク I/O が行われていることを示す。	通常、「ウェイクアップ」すると直ちに強制終了される。プロセスによってはウェイクアップしないものがあり、クリアするにはサーバの再起動が必要。
runnable	実行可能なプロセスのキュー内にある。	即時
running	サーバ・エンジンの 1 つで実行中。	即時
infected	サーバが重大なエラー状態を検出した。この状態が発生することはほとんどない。	kill コマンドの実行はすすめられない。プロセスをクリアするためにサーバを再起動しなければならない可能性が高い。
background	ユーザ・プロセスによってではなく Adaptive Server によって実行される、スレッショルド・プロシージャなどのプロセス。	即時終了。慎重に kill を使用すること。バックグラウンド・プロセスを強制終了する前に、sysprocesses を十分に確認することをすすめる。
log suspend	ログでラストチャンス・スレッショルドに達したために中断されているプロセス。	即時

kill コマンドを発行できるのはシステム管理者だけです。このコマンドの使用パーミッションを譲渡することはできません。

構文は次のとおりです。

kill *spid*

2 つ以上のプロセスを同時に強制終了することはできませんが、連続する kill コマンドをバッチとして実行できます。次に例を示します。

```
1> kill 7
2> kill 8
3> kill 9
4> go
```

kill コマンドの取り消しはできません。また、ユーザ定義のトランザクション内で実行することはできません。spid は数値定数です。変数は使用できません。次は、sp_who の出力の例を示しています。

```
fid spid status     loginame origname  hostname  blk  dbname cmd
--- ---- ---------  -------- --------  --------  ---  ------ ----------------
0   1    recv sleep  howard   howard    svr30eng  0    master AWAITING COMMAND
0   2    sleeping    NULL     NULL                0    master NETWORK HANDLER
0   3    sleeping    NULL     NULL                0    master DEADLOCK TUNE
0   4    sleeping    NULL     NULL                0    master MIRROR HANDLER
0   5    sleeping    NULL     NULL                0    master CHECKPOINT SLEEP
0   6    sleeping    NULL     NULL                0    master HOUSEKEEPER
0   7    recv sleep  bill     bill      bigblue   0    master AWAITING COMMAND
0   8    recv sleep  wilbur   wilbur    hazel     0    master AWAITING COMMAND
0   9    recv sleep  joan     joan      luv2work  0    master AWAITING COMMAND
0   10   running     foote    foote     svr47hum  0    master SELECT
(10 rows affected, return status = 0)
```

この例のプロセス 2 ～ 6 は、強制終了が不可能です。これらはシステム・プロセスです。ログイン名が NULL であることとホスト名がないことから、システム・プロセスであることがわかります。NETWORK HANDLER、MIRROR HANDLER、HOUSEKEEPER、および CHECKPOINT SLEEP (まれに CHECKPOINT のこともある) は常に sp_who 出力に表示されます。監査が使用可能であると、AUDIT PROCESS が表示されます。

プロセス 1、8、9、10 は、ステータス値が “recv sleep”、“send sleep”、“alarm sleep”、“lock sleep” であるので、強制終了が可能です。

sp_who の出力では、“recv sleep” が Adaptive Server ユーザのものでコマンドの結果を見るために一時停止させられているのか、あるいはユーザが PC などの端末を再起動したためにプロセスが中断状態で残っているのかはわかりません。このようなプロセスの情報を得るには、sysprocesses テーブルを問い合わせます。たとえば、次のクエリでは、プロセス 8 が使用しているホスト・プロセス ID とクライアント・ソフトウェアが表示されています。

```
select hostprocess, program_name
    from sysprocesses
where spid = 8

hostprocess program_name
----------- ----------------
3993        isql
```

このクエリと、sp_who の結果から得られたユーザおよびホストについての情報を参考にして、オペレーティング・システム・レベルでプロセスを追跡できます。

statusonly を指定した kill の使用

kill ...statusonly は、ロールバック・ステータスであるサーバ・プロセス ID (spid) の進捗状況についてレポートします。指定した spid は強制終了されません。statusonly レポートには、ロールバックの完了率と完了までにかかる推定時間 (秒単位) が表示されます。ロールバックの進捗状況を追跡するには、kill...with statusonly を複数回実行する必要があります。

```
kill spid with statusonly
```

spid は、停止するプロセスの番号です。

たとえば、次の例は spid 番号 13 のロールバック・プロセスについてレポートします。

```
kill 13 with statusonly
spid:13 Transaction rollback in progress.Estimated rollback completion:17% Estimated time
left:13 seconds
```

kill...statusonly を発行したときに、指定した spid のロールバックが既に完了している場合、または指定した spid がロールバックできない場合は、kill...statusonly から次のメッセージが返されます。

```
Status report cannot be obtained.KILL spid:nn is not in
progress.
```

sp_lock によるブロック・プロセスの調査

前述の sp_who の他に、システム・プロシージャ sp_lock も、他のプロセスをブロックしているプロセスの特定に利用できます。sp_who のレポートの blk_spid カラムに、別のプロセスがブロックされて、ロック取得のために待機中であることが示されている場合に、sp_lock を実行すると、ブロックしているプロセスについての情報を表示できます。たとえば、前述の sp_who の出力にあるプロセス 10 は、プロセス 7 によってブロックされています。プロセス 7 についての情報を表示するには、次を実行します。

```
sp_lock 7
```

Adaptive Server でのロッ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：ロックと同時実行制御』を参照してください。

ハウスキーピング機能

ハウスキーピング・タスクには、次の重要な機能があります。

- ハウスキーピングを構成するタスクには、ハウスキーピング・ウォッシュ ハウスキーピング・ガーベジ・コレクション、ハウスキーピング・チョアの 3 つがあります。次の出力に示されるように、sp_who はこの 3 つのタスクをすべて認識します。

```
fid    spid   status       loginame     orignname     hostname   blk_sp
       id       dbname          cmd     block_xloid
----  -----  ----------    ----------    ---------   ----------  ------
  -------    ---------  -------------  -------------
   0      5    sleeping          henry          NULL    luv2work       0
   master     tempdb          select             0
   0      6    sleeping            joe          NULL         NULL      0
   master     tempdb           HK GC             0
   0      7     sleeping          NULL         NULL         NULL       0
   master      tempdb        HK CHORES            0

(11 rows affected, return status = 0)
```

- ハウスキーピング関連のすべてのシステム・タスクを自動的に再起動します。これらのシステム・タスクが予期せず終了しても、サーバを再起動する必要はありません。
- システム管理者は、すべてのハウスキーピング・タスクの優先度を変更できます。

  sp_showpsexe も sp_who と同様に、3 つのハウスキーピングの名前をすべて認識します。

sp_who と sp_showpsexe 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。

ハウスキーピング・ウォッシュ

バッファのウォッシュは必須のタスクではなく、有効にした場合、アイドル時にのみ実行されます。このタスクを実行しないようにするには、housekeeper free write percent 設定パラメータを使用します。ハウスキーピング・タスクのうち、ハウスキーピング・ウォッシュ・タスクだけがこの設定パラメータを使用します。

ハウスキーピング・チョア

ハウスキーピング・チョア・タスクは、アイドル時にのみ実行されます。共通の設定パラメータはありません。このタスクは、次のような雑多な処理を管理します。

- テーブル統計情報をフラッシュする。
- アカウント統計情報をフラッシュする。
- 分散トランザクションのタイムアウトを処理する。この処理を行わないようにするには、dtm detach timeout period 設定パラメータを使用します。
- ライセンスの使用状況を調べる。この処理を行わないようにするには、license information 設定パラメータを使用します。

ハウスキーピング・ガーベジ・コレクション

ガーベジ・コレクションには、「消極的」と「積極的」の 2 つの形態があります。これらの名前は、空ページを探す 2 種類のテストの性質を表しています。

- 消極的ガーベジ・コレクションは、コストをかけずに空ページを探すテストを指します。このテストは、トランザクションが長時間実行されている間は効果がないことがあり、空ページが累積していく可能性があります。消極的ガーベジ・コレクションは低コストで実行できますが、パフォーマンスを低下させることがあります。パフォーマンスに影響を与えるものには、割り付けられたテーブル領域の断片化と、クエリの実行時に調べる必要がある空ページの累積があります。
- 積極的ガーベジ・コレクションは、より高度な方法で空ページを検索するテストを指します。このテストは、ページ内の削除されたローを 1 つずつ調べて、そのローを削除したトランザクションがコミットされているかどうかを判断するので、消極的ガーベジ・コレクション・テストよりもコストが高くなります。

  delete コマンドとハウスキーピング・ガーベジ・コレクション・タスクについて、積極的ガーベジ・コレクションを行うか消極的ガーベジ・コレクションを行うかを設定するには、enable housekeeper GC 設定パラメータを使用します。

  積極的ハウスキーピング・ガーベジ・コレクションの場合は、タスクによってハウスキーピング対象リストを検査する頻度は、アプリケーションによって空ページが生成される速さと一致するように自動的に調整されます。

ユーザの優先度での実行

ハウスキーピング・ガーベジ・コレクション・タスクは一般ユーザの優先度レベルで動作するので、CPU 時間に関して通常のユーザ・タスクと競合することになります。したがって、ハウスキーピングによる空きページの削除を上回る速さで、空きページのリストが増大することはありません。

enable housekeeper GC の設定

ガーベジ・コレクション・タスクに関して Adaptive Server を設定するときの構文は次のとおりです。

```
sp_configure "enable housekeeper GC", value
```

たとえば、次のように入力します。

```
sp_configure "enable housekeeper GC", 4
```

enable housekeeper GC 設定パラメータの有効な値は次のとおりです。

- 0 − ハウスキーピング・ガーベジ・コレクション・タスクは実行しませんが、delete コマンドによる消極的ガーベジ・コレクションは実行できるようにします。reorg reclaim_space を使用して、空ページの割り付けを解除する必要があります。これは、パフォーマンスへの影響が最も少なく、最も低コストのオプションですが、累積した空ページの量が増えるとパフォーマンス上の問題が発生する可能性があります。この値を使用することはおすすめしません。
- 1 − ハウスキーピング・ガーベジ・コレクション・タスクと delete コマンドの両方で、消極的ガーベジ・コレクションを実行できます。これはデフォルトの値です。アプリケーションで許容される以上の空ページが累積する場合は、オプション 4 または 5 の使用を検討してください。optdiag ユーティリティを使用すると、空ページの統計情報を取得できます。
- 2 − 今後のために予約済み。
- 3 − 今後のために予約済み。
- 4 − ハウスキーピング・ガーベジ・コレクション・タスクと delete コマンドの両方で、積極的ガーベジ・コレクションを実行できます。このオプションを選択すれば効果が最も高くなりますが、delete コマンドのコストは最も高くなります。このオプションは、データオンリーロック・テーブルに対する一連の削除を 1 つのバッチで実行する場合に理想的です。
- 5 − ハウスキーピング・タスクでは積極的ガーベジ・コレクションを実行でき、delete では消極的ガーベジ・コレクションを実行できます。オプション 4 を選択した場合よりも、削除のコストは低くなります。このオプションは、同時トランザクションによって削除が行われる場合に適しています。

reorg コマンドの使用方法

ガーベジ・コレクションの効果が最も高くなるのは、enable housekeeper GC を 4 または 5 に設定した場合です。このパラメータ値を 5 に設定することをおすすめしますが、パフォーマンスを考慮する上でこのパラメータを 4 または 5 に設定できないこともあります。その場合は、空ページが累積したときに、影響を受けているテーブルに対して reorg を実行してください。空ページに関する統計情報を取得するには、optdiag ユーティリティを使用します。

サーバが停止またはクラッシュすると、ハウスキーピング・ガーベジ・コレクション・タスクがまだ処理していないページ割り付け解除要求は失われます。このような、空ではあるけれどもハウスキーピング・ガーベジ・コレクション・タスクが割り付けを解除していないページは、reorg によって削除されるまでは、割り付け済みのままになります。

『システム管理ガイド 第 2 巻』の「第 9 章 reorg コマンドの使用方法」を参照してください。

SQL バッチ・テキストを保存するための Adaptive Server の設定

場合によっては、クエリまたはプロシージャによって Adaptive Server Monitor は応答が停止することがあります。システム管理者の役割を持つユーザは、現在実行されている SQL バッチのテキストに Adaptive Server Monitor がアクセスできるように Adaptive Server を設定することができます。実行時間の長いバッチの SQL テキストを表示することによって、「スタックした」プロセスをデバッグしたり、リソースを大量に消費する長い文を細かく調整したりすることができます。

SQL バッチ・テキストを収集して共有メモリに書き込むように Adaptive Server を設定する必要があります。この共有メモリから、Adaptive Server Monitor Server (Adaptive Server Monitor のサーバ・コンポーネント) がテキストを読み込みます。クライアントの要求は、Sybase Central のプラグインである Monitor Viewer、または他の Adaptive Server Monitor Server アプリケーションから発行できます。

SQL バッチ・テキストを保存するように Adaptive Server を設定した場合は、現在のクエリ・プランを showplan フォーマット (showplan をオンに設定したときに表示されるものと同じ) で表示することもできます。現在のクエリ・プランを Adaptive Server で表示する方法については、「SQL 文のクエリ・プランの表示」(358 ページ) を参照してください。SQL バッチは、Adaptive Server Monitor Server を通してのみ表示が可能です。バッチ・テキストの表示の詳細については、Adaptive Server Monitor Server のマニュアルを参照してください。

表示しようとしているクエリまたはプロシージャは、SQL テキスト内でネストしていることもあります。このため、sysprocesses テーブルには、応答を停止している文のクエリ・プランを分析できるように、行番号、文番号、spid のカラムが追加されています。

デフォルトでは、Adaptive Server は SQL バッチ・テキストを保存するように設定されていません。したがって、この機能のためのメモリを割り付けるようにシステム管理者が Adaptive Server を設定する必要があります。SQL バッチを保存するためのメモリを設定しなければ、Adaptive Server Monitor から SQL にアクセスすることによるパフォーマンスへの影響はありません。

バッチ・テキストへのメモリの割り付け

保存する SQL バッチ・テキストの量を設定します。テキストが保存されるように設定すると、それ以降の SQL テキスト・バッチは、SQL Server Monitor との共有メモリにコピーされます。新しいバッチが発生するたびにその接続のメモリがクリアされ、直前のバッチは上書きされるので、表示できるのは現在実行している SQL 文だけです。

❖ SQL テキストの保存

1 メモリに保持する SQL テキストの量を設定します (「メモリに保持する SQL テキストの量の設定」(356 ページ) 参照)。

2 SQL テキストの保存を開始するように Adaptive Server を設定します (「SQL テキストの保存を開始するための Adaptive Server の設定」(357 ページ) 参照)。

注意 SQL テキスト・バッチの設定と保存を行うには、システム管理者の権限が必要です。

メモリに保持する SQL テキストの量の設定

インストール後、共有メモリにコピーされる SQL テキストの最大量を決定する必要があります。ユーザ 1 人あたりに割り付けるメモリ量を決めるには、以下を参照してください。

- 割り付けられた量のメモリを超えた SQL バッチは、警告が表示されないままトランケートされる。つまり、バッチ文用に十分なメモリを割り付けないと、表示しようとしているテキストが、トランケートされたバッチの部分に含まれることがあります。
- SQL テキスト用に共有メモリから割り付けるメモリ量を増やせば、共有メモリにコピーされるバッチから目的の文がトランケートされる確率は低くなります。ただし、非常に大きな値を指定すると、データやプロシージャのキャッシュ用に十分なメモリが残らなくなるので、そのような指定は Adaptive Server によって即座に拒否されます。

Sybase では、1 ユーザ接続あたりの初期値を 1,024 バイトにすることをおすすめします。

共有メモリを割り付けるには、sp_configure を使用して max SQL text monitored 設定パラメータを設定します。ここで、*bytes_per_connection* (クライアント接続当たりの最大保存バイト数) は 0 (デフォルト) ～ 2,147,483,647 (理論上の上限) です。

```
sp_configure "max SQL text monitored", bytes_per_connection
```

このパラメータを有効にするには、Adaptive Server の再起動が必要です。

SQL テキスト用に割り付けられている共有メモリの合計メモリ量は、*bytes_per_connection* とユーザ接続数を乗算した値です。

SQL テキストの保存を開始するための Adaptive Server の設定

SQL テキスト用に共有メモリを割り付けた後は、SQL バッチが含まれる Adaptive Server Monitor のイベント・サマリを有効にするたびに各 SQL バッチのコピーが保存されます。

また、Adaptive Server Monitor がイベント・バッファをスキャンして SQL テキストがあるかどうかを調べる間隔も再設定する必要があります。Adaptive Server Monitor のマニュアルを参照してください。

テキストで表されない SQL コマンド

テキストで表されない Client-Library 関数 (ct_cursor や ct_dynamic など) を使用して SQL コマンドが発行されたときは、Client-Library は効率向上のため情報をコード化し、Adaptive Server は主要なコマンド情報を復号化して表示します。たとえば、ct_cursor でカーソルをオープンした場合に、そのコマンドが実行されているときは、Adaptive Server Monitor のイベント・サマリにはカーソル名とカーソル宣言文が表示されます。

表 11-3 は、テキストで表されない Client-Library 関数の一覧です。

表 *11-3:* テキストで表されない *SQL* コマンド

Client-Library ルーチン	DB-Library ルーチン	表示名	表示データ
ct_cursor	該当なし	CLOSE_CURSOR	カーソル名、文
ct_cursor	該当なし	DECLARE_CURSOR	カーソル名、文
ct_cursor	該当なし	DELETE_AT_CURSOR	カーソル名、文
ct_cursor	該当なし	FETCH_CURSOR	カーソル名、文
ct_fetch (ct_cursor の結果の処理時)	該当なし	FETCH_CURSOR	カーソル名、文
ct_cursor CURSOR_ROWS、接続に Client-Library カーソルがある場合は ct_cancel	該当なし	CURSOR_INFO	カーソル名、文
ct_cursor	該当なし	OPEN_CURSOR	カーソル名、文
ct_cursor	該当なし	UPDATE_AT_CURSOR	カーソル名、文
ct_command (CS_RPC_CMD) (デフォルト動作)	dbrpcinit (バージョン 10.0.1 以降のみ)	DBLIB_RPC	RPC 名
ct_dynamic	該当なし	DYNAMIC_SQL	動的文の名前、文
ct_command (CS_MSG_CMD)	該当なし	MESSAGE	なし
ct_param	dbrpcparam	PARAM_FORMAT	なし
ct_param	dbrpcparam	PARAMS	なし
ct_command (CS_RPC_CMD) (バージョンが 5.0 より前の TDS を使用している場合)	dbrpcparam (バージョン 10.0.1 以前の DB-Library)	RPC	RPC 名

テキストで表されない SQL コマンドの詳細については、Open Client のマニュアルを参照してください。

SQL 文のクエリ・プランの表示

sp_showplan と、調査対象のユーザ接続の *spid* を使用して、その接続で現在実行されている文のクエリ・プランを取り出します。また、sp_showplan を使用して、同じバッチ内の以前の文のクエリ・プランを表示することもできます。

```
declare @batch int
declare @context int
declare @statement int
execute sp_showplan <spid_value>, @batch_id= @batch output,
@context_id= @context output, @stmt_num=@statement output
```

構文の説明は、次のとおりです。

- *batch_id* − バッチのユニークな番号
- *context_id* − そのバッチ内で実行される個々のプロシージャ (またはトリガ) のユニークな番号
- *stmt_num* − バッチ内の現在の文の番号

Adaptive Server は、ユニークなバッチ ID を使用して、バッチ・テキストなどの Adaptive Server Monitor によって取り出されるデータとクエリ・プランとを同期させます。

注意 sp_showplan を実行するにはシステム管理者の権限が必要です。

たとえば、spid 99 の現在の文のクエリ・プランを表示するには、次のように入力します。

```
declare @batch int
declare @context int
declare @statement int
exec sp_showplan 99, @batch output, @context output, @statement output
```

クエリ・プランのプロシージャは、Adaptive Server が SQL テキスト用に共有メモリを割り付けているかどうかにかかわらず、Adaptive Server Monitor とは独立して実行できます。

以前の文の表示

同じバッチ内にある以前の文のクエリ・プランを表示するには、元のクエリと同じ値 (ただし文番号は 1 を引いた値) を指定して、sp_showplan を発行します。この方法を使用すると、クエリ番号 1 番までさかのぼってバッチ内のすべての文を表示することができます。

ネストしているプロシージャの表示

sp_showplan を使用すると現在の文のクエリ・プランを表示できますが、実際に表示されている文は、元の SQL バッチから呼び出されたプロシージャ内 (またはネストしたプロシージャのチェーン内) に存在していることがあります。表 11-4 に、このようなネストされた文に関する情報が保管されている sysprocesses のカラムを示します。

表 *11-4:* ネストされた文の *sysprocesses* カラム

カラム	データ型	内容
id	整数	実行されているプロシージャのオブジェクト ID (プロシージャが実行されていない場合は 0)。
stmtnum	整数	実行されているプロシージャ内の現在の文番号 (プロシージャが実行されていない場合は、SQL バッチ文の番号)。
linenum	整数	実行されているストアド・プロシージャ内の現在の文の行番号 (プロシージャが実行されていない場合は、現在の SQL バッチ文の行番号)。

SQL テキストが有効になっているか、または SQL テキスト用にメモリが割り付けられているかどうかにかかわらず、この情報は sysprocesses に保存されます。

id カラム、stmtnum カラム、linenum カラムを表示するには、次のように入力します。

```
select id, stmtnum, linenum
from sysprocesses
where spid = spid_of_hung_session
```

注意　この select 文を実行するために sa_role は必要ありません。

サーバの停止

システム管理者は、Adaptive Server または Backup Server を停止させることができます。使用する構文は次のとおりです。

shutdown [*backup_server_name*] [with {wait|nowait}]

shutdown コマンドのデフォルトは with wait です。したがって、shutdown と shutdown with wait の結果はまったく同じです。

Adaptive Server の停止

サーバ名を指定しないで shutdown を実行した場合は、使用中の Adaptive Server が停止します。shutdown コマンドが発行されると、Adaptive Server は次の処理を行います。

1 システム管理者以外はログインできないようにします。

2 個々のデータベースでチェックポイントを実行して、変更されたページをメモリからディスクにフラッシュします。

3 現在実行中の SQL 文やプロシージャの終了を待ちます。

このようにして、shutdown は Adaptive Server の再起動時に自動リカバリが行わなければならない作業量を最小にします。

with nowait オプションを指定した場合は、Adaptive Server はただちに停止します。ユーザ・プロセスはアボートされ、shutdown with nowait の後のリカバリの時間は長くなります。shutdown with nowait コマンドを発行する前に checkpoint コマンドを発行すると、リカバリ時間を短縮するのに役立ちます。

Backup Server の停止

Backup Server を停止するには、その Backup Server の名前を含めます。

```
shutdown SYB_BACKUP
```

デフォルトは with wait であるため、進行中のすべてのダンプやロードが終了してから Backup Server のプロセスが停止します。shutdown コマンドが発行された後は、その Backup Server で新しいダンプ・セッションやロード・セッションを開始することはできません。

使用している Adaptive Server からアクセスできる Backup Server の名前を調べるには、sp_helpserver を実行します。name カラムの値を shutdown コマンドで使用してください。Backup Server を停止できるのは、次の状態のときだけです。

- 使用している Adaptive Server の sysservers にリストされている
- 実行するユーザのローカルの *interfaces* ファイルにリストされている

Backup Server を sysservers に追加するには、sp_addserver を使用します。

アクティブなダンプおよびロードのチェック

shutdown コマンドを実行する前に Backup Server 上のアクティビティを確認するには、その Backup Server 上で sp_who コマンドを実行します。

```
SYB_BACKUP...sp_who

spid   status   loginame hostname   blk cmd
------ -------- -------- ---------- --- --------------
   1 sleeping NULL     NULL       0   CONNECT HANDLER
   2 sleeping NULL     NULL       0   DEFERRED HANDLER
   3 runnable NULL     NULL       0   SCHEDULER
   4 runnable NULL     NULL       0   SITE HANDLER
   5 running  sa       heliotrope 0   NULL
```

Backup Server での *nowait* の使用

shutdown *backup_server* with nowait コマンドは、現在のアクティビティに関係なく Backup Server を停止します。これは、重大な問題が発生した場合にだけ使用してください。これを使用すると、ダンプまたはロードが不完全になったり、一貫性のない状態で残ることがあります。

ログまたはデータベースのダンプの実行中に shutdown with nowait を使用する場合は、ダンプ終了を示すメッセージを確認してください。このメッセージを受け取っていない場合、またはダンプが終了したかどうかが確かでない場合は、次回のダンプではトランザクション・ダンプではなく、dump database を使用してください。このようにすれば、信頼すべきダンプの一貫性が失われているという疑いを持たなくてすみます。

shutdown with nowait を実行したときに何らかのロードが実行中であり、ロードの終了を示すメッセージを受け取っていない場合は、そのデータベースに対して load transaction コマンドを発行できなくなることがあります。そのデータベースを使用する前に、完全なデータベース一貫性チェック (dbcc) を実行してください。load database から始まる一連のロード・コマンドをすべて発行し直さなければならない場合があります。

既知の問題についての情報

『リリース・ノート』には、Adaptive Server と Backup Server に関する既知の問題や互換性についての重要な情報があります。あらかじめ『リリース・ノート』を参照しておくことによって、このような問題の解決に費やす時間を節約し、無用な推測を避けることができます。

パート 2

セキュリティの管理

次の各章では、Adaptive Server のセキュリティの管理について説明します。

- 「第 12 章 セキュリティの概要」では、セキュリティの概念を紹介します。
- 「第 13 章 Adaptive Server のセキュリティ管理について」では、Adaptive Server で使用できるセキュリティ機能の概要について説明します。
- 「第 14 章 Adaptive Server のログイン、データベース・ユーザ、クライアント接続の管理」では、Adaptive Server のログイン・アカウントとデータベース・ユーザを管理する方法について説明します。
- 「第 15 章 リモート・サーバの管理」では、各 Adaptive Server のシステム管理者とシステム・セキュリティ担当者が、リモート・プロシージャ・コール (RPC) を使用できるようにするために実行する手順について説明します。
- 「第 16 章 外部認証」では、ユーザの認証とネットワークを介して転送するデータの保護を可能にする、ネットワーク・ベースのセキュリティ・サービスについて説明します。
- 「第 17 章 ユーザ・パーミッションの管理」では、ユーザ・パーミッションの使用と実装について説明します。
- 「第 18 章 監査」では、インストール環境に応じた監査の設定方法について説明します。
- 「第 19 章 データの機密保持」では、すべてのデータを保護し、機密性を保持するための Adaptive Server の設定方法について説明します。

第 12 章

セキュリティの概要

セキュリティの概要

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

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

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

情報セキュリティの概要

次に、企業のセキュリティを考慮する際の一般的なガイドラインを示します。

- 重要な情報の機密性が保たれていること – 誰がどの情報にアクセスできるかを決めます。
- システムの整合性が保たれていること – サーバは、ルールと制約を使用して、情報の正確性と完全性が保たれることを保証する必要があります。
- 必要な情報が入手可能であること – すべての保護手段が機能している場合でも、情報にアクセスする人物が必要に応じて情報を入手できる必要があります。

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

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

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

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

Adaptive Server には、企業のセキュリティ・ポリシーを強制するのに役立つ一連のセキュリティ機能が含まれています。Adaptive Server のセキュリティ機能の詳細については、「第 13 章 Adaptive Server のセキュリティ管理について」を参照してください。

情報セキュリティ規格

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

この項では、これらの認定について説明します。

Common Criteria 設定評価

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

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

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

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

FIPS 140-2 検証済み暗号化モジュール

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

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

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

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

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

第 13 章

Adaptive Server のセキュリティ管理について

セキュリティ管理の一般処理

表 13-1 は、Adaptive Server のセキュリティ管理に必要な主要タスクの説明と、各タスクの実行方法についての指示が記載されているマニュアルを示します。

表 *13-1:* セキュリティ管理の一般処理

作業	説明	参照箇所
1. 監査機能を含む、Adaptive Server のインストール	この作業には、インストールの準備、配布メディアからのファイルのロード、実際のインストール、必要な物理リソースの管理が含まれる。	使用しているプラットフォームの『インストール・ガイド』および「第 18 章 監査」
2. 安全な管理環境の設定	これには、監査の有効化、各ユーザの責任を徹底させるためのユーザへの役割の付与、システム管理者とシステム・セキュリティ担当者へのログイン名の割り当て、パスワードとログイン・ポリシーの設定が含まれる。	「第 14 章 Adaptive Server のログイン、データベース・ユーザ、クライアント接続の管理」

作業	説明	参照箇所
3. サーバへのユーザ・ログインの追加、データベースへのユーザの追加、グループと役割の確立、代理認証の設定	ログインの追加、グループの作成、データベースへのユーザの追加、ログインの削除とロック、初期パスワードの割り当てが含まれる。ユーザへの役割の割り当て、ユーザ定義の役割の作成、役割階層と役割の相互排他の定義を行う。	「第 14 章 Adaptive Server のログイン、データベース・ユーザ、クライアント接続の管理」
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。詳細なアクセス制御を実施するためのアクセス・ルールを作成する。	「第 17 章 ユーザ・パーミッションの管理」
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する。機密データの暗号化。	カラム・レベルでの暗号化を使用し、暗号化するデータ・カラムを決定し、一度でキー作成を行う操作を実行し、alter table を使用して初期データ暗号化を実行するために、Adaptive Server を設定する。	『暗号化カラム・ユーザーズ・ガイド』
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、参照制約を追加する。	『ASE Transact-SQL ユーザーズ・ガイド』および『リファレンス・マニュアル：コマンド』
7. 監査の設定と管理	監査対象を決定し、Adaptive Server の使用を監査する。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する。	「第 18 章 監査」と、使用しているプラットフォームの『インストール・ガイド』と『設定ガイド』
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する。	「第 16 章 外部認証」および「第 19 章 データの機密保持」

セキュリティの設定に関する推奨事項

次に、ログインとセキュリティの関連について説明します。

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

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

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

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

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

セキュリティの設定例

表 13-2 に示すユーザに特別な役割を割り当てる場合を想定します。

表 *13-2:* 役割を割り当てる予定のユーザ

名前	権限	オペレーティング・システムのログイン名
Rajnish Smith	sso_role	rsmith
Catherine Macar-Swan	sa_role	cmacar
Soshi Ikedo	sa_role	sikedo
Julio Rozanski	oper_role	jrozan
Alan Johnson	dbo	ajohnson

表 13-3 は、表 13-2 に示した役割の割り当てに基づいて Adaptive Server の安全な操作環境を設定するために使用する一連のコマンドを示します。オペレーティング・システムにログインしたら、初期設定されている “sa” アカウントを使用して次のコマンドを発行します。

表 *13-3:* セキュリティの設定に使用するコマンドの例

コマンド	結果
• isql -Usa	“sa” として Adaptive Server にログインする。sa_role と sso_role の両方がアクティブである。
• sp_audit “security”,“all”, “all”, “on” • sp_audit “all”, “sa_role”, “all”, “on” • sp_audit “all”, “sso_role”, “all”, “on”	サーバ全体のセキュリティ関連イベントに対する監査オプションを設定する。また、sa_role または sso_role がアクティブなすべてのアクションの監査を設定する。
• sp_configure “auditing” 1	監査を有効にする。

注意 監査証跡用のスレッショルド・プロシージャを設定し、sybsecurity でのトランザクション・ログの処理方法を決定してから、監査を有効化すること。「第 18 章 監査」を参照してください。

• sp_addlogin rsmith, js&2P3d, @fullname = “Rajnish Smith” • sp_addlogin cmacar, Fr3ds#1, @fullname = “Catherine Macar-Swan” • sp_addlogin sikedo, mi5pd1s, @fullname = “Soshi Ikedo” • sp_addlogin jrozan, w1seCrkr, @fullname = “Julio Rozanski”	Rajnish、Catherine、Soshi、Julio のログインとパスワードを追加する。 どのユーザにもデフォルト・データベースは指定していないので、これらのユーザのデフォルト・データベースは master になる。
• grant role sso_role to rsmith • grant role sa_role to sikedo • grant role sa_role to cmacar • grant role oper_role to jrozan	Rajnish に sso_role、Soshi と Catherine に sa_role、Julio に oper_role 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。
use master sp_addlogin ajohnson, j06n50n, @fullname = “Alan Johnson” create database sales_summary use sales_summary sp_changedbowner ajohnson sp_modifylogin ajohnson, 'defdb', sales_summary	新しいデータベース sales_summary を作成し、Alan をこのデータベースの所有者にする。Alan はデータベース所有者であるため、このデータベース内でユーザの作成、新しいデータベース・オブジェクトの作成、他のユーザへのパーミッションの付与を行うことができる。
sp_locklogin sa,“lock”	他人が “sa” としてログインできないよう、“sa” ログインをロックする。各ユーザは、各自に設定された役割だけを使用できる。

注意 個々のユーザに sa_role と sso_role の各役割を付与し、これらの役割が正常に機能することを確認してから、“sa” ログインをロックすること。

Adaptive Server のセキュリティ機能

表 13-4 は、Adaptive Server のセキュリティ機能を示します。

表 *13-4:* 主要なセキュリティ機能

セキュリティ機能	説明
識別と認証の制御	承認されたユーザだけがシステムにログインできるようにする。Adaptive Server は、パスワードベースのログイン認証の他に、Kerberos、LDAP、PAM による外部認証もサポートしている。
任意アクセス制御 (DAC)	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機能。通常は grant コマンドと revoke コマンドを使用する。この種の制御は、オブジェクトの所有者が自由に設定できる。
役割の分担	権限が付与された役割を複数の指定ユーザに割り当てて、指定ユーザだけが特定のタスクを実行できるようにする。Adaptive Server には、システム管理者やシステム・セキュリティ担当者などの「システム標準の役割」と呼ばれる、事前に定義された役割がある。また、システム・セキュリティ担当者が「ユーザ定義の役割」と呼ばれる追加の役割を定義できる。
責任範囲の明確化のための監査	ログイン、ログアウト、サーバの起動操作、リモート・プロシージャ・コール、データベース・オブジェクトへのアクセス、特定ユーザによってまたは特定の役割をアクティブにして実行されたすべてのアクションなどのイベントを監査する機能。1 つのオプションを設定するだけで、サーバ全体にわたる一連のセキュリティ関連イベントを監査することもできる。
データの機密保持	クライアント／サーバ間の通信に Kerberos や SSL による暗号化を使用して、データの機密性を保持する。カラム・レベルの暗号化では、データベースに保存されたデータの機密性を保持する。アクティブでないデータは、パスワードで保護されたデータベース・バックアップによって機密性を保持される。

識別と認証

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

識別と認証の詳細については、「第 14 章 Adaptive Server のログイン、データベース・ユーザ、クライアント接続の管理」を参照してください。また、「代理権限の使用」(561 ページ) と「第 15 章 リモート・サーバの管理」も参照してください。

外部認証

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

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

これらの外部認証方式の詳細については、「第 16 章 外部認証」を参照してください。

リモート・サーバの管理

Adaptive Server 間でログインとユーザを管理する内部メカニズムについては、「第 15 章 リモート・サーバの管理」を参照してください。

任意アクセス制御

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

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

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

任意アクセス制御については、「第 17 章 ユーザ・パーミッションの管理」を参照してください。

ロー・レベル・アクセス制御

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

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

- アクセス・ルール
- Application context facility
- ログイン・トリガ
- ドメイン整合性ルール

「ロー・レベル・アクセス制御の使用」(577 ページ) を参照してください。

役割の分担

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

これらの役割により、システムの操作と管理作業を実行するユーザの責任が明確になります。そして、作業を監査し、どのユーザの作業かを明確にできます。

役割の階層

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

相互排他性

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

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

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

「ユーザに対する役割の作成と割り当て」(390 ページ) を参照してください。

責任範囲の明確化のための監査

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

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

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

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

- サーバ全体にわたるセキュリティ関連イベント
- データベース・オブジェクトの作成、削除、変更
- 特定ユーザが行ったすべてのアクション、または特定の役割をアクティブにしてユーザが行ったすべてのアクション
- データベース・アクセス権の付与または取り消し

- データのインポートまたはエクスポート
- ログインとログアウト
- 暗号化キーに関連するすべての作業

監査機能については、「第 18 章 監査」を参照してください。

データの機密保持

Adaptive server では、SSL (Secure Socket Layer) 標準や Kerberos を使用してクライアント・サーバ間の通信を暗号化することにより、データの機密性を保持できます。データベース内でカラム・レベルの暗号化を行い、オフライン・データのバックアップを暗号化することにより、データの機密性を保護することができます。

詳細については、次の情報を参照してください。

- SSL － 「第 19 章 データの機密保持」
- Kerberos － 「第 16 章 外部認証」
- 暗号化カラム － 『暗号化カラム・ユーザーズ・ガイド』

パスワードで保護されたデータベース・バックアップ

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。『リファレンス・マニュアル：コマンド』と『システム管理ガイド 第 2 巻』の「第 12 章 ユーザ・データベースのバックアップとリストア」を参照してください。

第 14 章

Adaptive Server のログイン、データベース・ユーザ、クライアント接続の管理

Adaptive Server への新しいログインの追加、データベースへのユーザの追加、コマンドやデータベース・オブジェクトを使用するための「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。

これらの手順では、sp_addlogin を使用して、特定のサーバに対するログイン・アカウントを作成します。このとき、そのサーバの *syslogins* テーブルにアカウント情報が保存されます。別の方法として、LDAP サーバにログイン・アカウントを作成して格納することもできます。

1 システム・セキュリティ担当者が、sp_addlogin を使用して新しいユーザのサーバ・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、sp_adduser を使用して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。「グループの作成」(383 ページ) を参照してください。エイリアスを使用してデータベースへのユーザ・アクセスを与えることができます。「エイリアスの追加」(411 ページ) を参照してください。

3 システム・セキュリティ担当者が、このユーザに特定の役割を付与します。

4 システム管理者、データベース所有者、またはオブジェクト所有者が、特定のコマンドとデータベース・オブジェクトに対するパーミッションを、ユーザまたはグループに付与します。オブジェクトに対する特定のパーミッションを別のユーザまたはグループに付与するパーミッションを、ユーザやグループに付与することもできます。「第 17 章 ユーザ・パーミッションの管理」を参照してください。

表 14-1 は、これらのタスクに使用するシステム・プロシージャとコマンドをまとめたものです。

表 14-1: Adaptive Server とデータベースへのユーザの追加

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
新しいログインの作成、パスワード、デフォルト・データベース、デフォルト言語、フルネームの割り当て	システム・セキュリティ担当者	sp_addlogin	任意のデータベース
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 および grant role	マスタ・データベース
データベースへのユーザの追加、グループの割り当て	データベース所有者またはシステム管理者	sp_adduser	ユーザ・データベース
その他のデータベース・ユーザに対するエイリアス・ユーザ	データベース所有者またはシステム管理者	sp_addalias	ユーザ・データベース
グループ、ユーザ、役割に対する、データベース・オブジェクトの作成パーミッションまたはアクセス・パーミッション、およびコマンドの実行パーミッションの付与	データベース所有者、システム管理者、システム・セキュリティ担当者、またはオブジェクト所有者	grant	ユーザ・データベース

パスワードの選択と作成

システム・セキュリティ担当者は、ユーザを Adaptive Server へのログインに追加するときに、各ユーザにパスワードを割り当てます。sp_password を使用すると、ユーザがいつでも自分のパスワードを変更できます。「パスワードの変更」(406 ページ) を参照してください。

パスワードを作成するときは、次の規則に従います。

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

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

次にパスワードの規則を示します。

- パスワードの長さは、6 文字以上でなければならない。
- 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、sp_addlogin で指定するときにパスワードを引用符で囲む。
  - A ～ Z、a ～ z、0 ～ 9、_、#、有効な 1 バイトまたはマルチバイトのアルファベット文字以外の文字を含む場合、またはアクセント付きのアルファベット文字を含む場合。
  - 0 ～ 9 の数字で始まる場合。

「複雑なパスワード・チェック」(427 ページ) を参照してください。

Adaptive Server へのログインの追加

Adaptive Server に新しい「ログイン」名を追加するには、sp_addlogin を使用します (sp_adduser を使用して、ユーザ・データベースにアクセスするためのパーミッションを付与します)。sp_addlogin 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。

sp_addlogin 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。

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、フルネームなしで設定します。

```
sp_addlogin "maryd", "100cents"
```

パスワードは 1 で始まるので、引用符が必要です。

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

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

```
sp_addlogin omar_khayyam, rubaiyat, pubs2
```

ユーザのフルネームを指定し、デフォルトのデータベースと言語を使用する場合は、*defdb* パラメータと *deflanguage* パラメータの代わりに null を指定してください。次に例を示します。

```
sp_addlogin omar, rubaiyat, null, null,
    "Omar Khayyam"
```

または、パラメータ名を指定することもできます。この場合はすべてのパラメータを指定する必要はありません。次に例を示します。

```
sp_addlogin omar, rubaiyat,
    @fullname = "Omar Khayyam"
```

sp_addlogin が実行されると、Adaptive Server は master.dbo.syslogins にローを追加し、新しいユーザにユニークなシステム・ユーザ ID (suid) を割り当て、その他の情報も記録します。ユーザがログインするとき、Adaptive Server はそのユーザが指定した名前とパスワードを syslogins の中で検索します。password カラムは一方向アルゴリズムで暗号化されるので、解読することはできません。

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

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

失敗したログイン

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

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

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

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

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

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

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

「ログイン失敗の監査」を参照してください。

グループの作成

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

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

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

```
sp_addgroup grpname
```

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

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

```
sp_addgroup senioreng
```

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

データベースへのユーザの追加

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

```
sp_adduser loginame [, name_in_db [, grpname]]
```

各パラメータの意味は、次のとおりです。

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

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

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

  注意 この機能は、「データベース内でのエイリアスの使用」(410 ページ) で説明するエイリアス機能とは異なります。エイリアスは、1 人のユーザの識別子とパーミッションを別の名前に対応付けるためのものです。

- *grpname* は、データベース内の既存のグループの名前です。グループ名を指定しない場合、そのユーザはデフォルト・グループ “public” のメンバになります。ユーザは、別のグループのメンバになっても、“public” グループのメンバであることに変わりはありません。「ユーザのグループ・メンバシップの変更」(408 ページ) を参照してください。

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

- use *database_name* コマンドを発行して、そのデータベースにアクセスする。
- sp_addlogin 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。
- sp_modifylogin を使用して、そのデータベースをデフォルトにする。

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

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

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

```
sp_adduser maryd
```

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

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

データベースへのアクセス権を持つユーザであっても、データベース内でのデータの読み込み、データの変更、特定のコマンドの使用といった操作を実行するには、パーミッションが必要です。このようなパーミッションを付与するには、grant コマンドと revoke コマンドを使用します。これらのコマンドについては、「第 17 章 ユーザ・パーミッションの管理」を参照してください。

“guest” ユーザのデータベースへの追加

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

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

```
sp_adduser guest
```

guest ユーザを削除するには sp_dropuser を使用します。詳細については、「ユーザの削除」(402 ページ) を参照してください。

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

注意　1 つのデータベースで複数のユーザが guest ユーザになることができますが、このときも、Adaptive Server はサーバ内でユニークな、ユーザのサーバ・ユーザ ID を使用して、各ユーザの実行記録を監査できます。「第 18 章 監査」を参照してください。

“guest” ユーザのパーミッション

“guest” は “public” の権限を継承します。データベース所有者とデータベース・オブジェクトの所有者は、grant と revoke を使用して、“guest” の権限を “public” の権限よりも拡大あるいは縮小することができます。「第 17 章 ユーザ・パーミッションの管理」を参照してください。

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

ユーザ・データベースの “guest” ユーザ

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

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

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

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

インストールされているシステム・データベースの “guest” ユーザ

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

pubs2 と *pubs3* の “guest” ユーザ

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

- すべてのユーザ・テーブルに対する select パーミッションとデータ変更パーミッション
- すべてのプロシージャに対する execute パーミッション
- create table、create view、create rule、create default、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、sp_addlogin を使用して、一時的に使用するユーザ (たとえば visitor) が使用するログイン名とパスワードを追加できます。通常、こうしたユーザには制限されたパーミッションを付与します。デフォルト・データベースを割り当てることもあります。

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

---

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

リモート・ユーザの追加

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

リモート・プロシージャ・コールを使用できるようにするには、ローカル・サーバとリモート・サーバの両方を設定する必要があります。「第 15 章 リモート・サーバの管理」を参照してください。

ユーザ ID とログイン ID の番号

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

ID 番号の制限と範囲

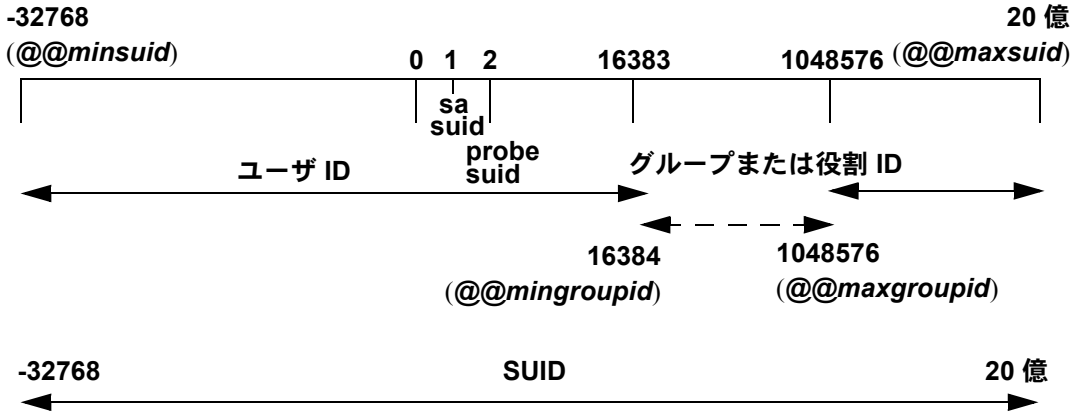
表 14-2 は、ID タイプごとの有効な範囲を示します。

表 *14-2: ID タイプの範囲*

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

図 14-1 は、ログイン、ユーザ、グループの制限と範囲を示します。

図 *14-1: Adaptive Server* で使用できるユーザ、グループ、ログイン



ユーザ ID (*uid*) に負の値が使用されることがあります。

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

ログイン接続の制限

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

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

注意 サーバ上で同時に実行されるタスクの最大数は 32,000 です。

❖ ログインと同時接続の数を最大にするには

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

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

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

表 14-3 は、ログイン、ユーザ、グループのサーバ制限に関するグローバル変数のリストです。

表 *14-3:* ログイン、ユーザ、グループに関するグローバル変数

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

グローバル変数を表示するには、次のように入力します。

```
select variable_name
```

次に例を示します。

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

ユーザに対する役割の作成と割り当て

データベース・ユーザ追加の手順の最後に、必要に応じてユーザに特別な役割を割り当て、パーミッションを付与します。パーミッションの詳細については、「第 17 章 ユーザ・パーミッションの管理」を参照してください。

Adaptive Server でサポートされる役割を使用すると、各ユーザの責任範囲を指定することができます。Adaptive Server には、システム管理者やシステム・セキュリティ担当者などのシステム標準の役割と、システム・セキュリティ担当者が作成し、ユーザや他の役割に付与された役割であるユーザ定義の役割があります。オブジェクト所有者は、必要に応じて、各役割にデータベース・アクセス権を付与できます。

システム標準の役割

表 14-4 は、システム標準の役割、grant role コマンドまたは revoke role コマンドの *role_granted* オプションに使用する値、その役割を持つユーザによって一般に実行されるタスクを示します。

表 *14-4:* システム標準の役割と関連するタスク

役割	*role_granted* の値	説明
システム管理者	sa_role	Adaptive Server のデータベースとディスク記憶領域の管理と維持
システム・セキュリティ担当者	sso_role	セキュリティ関連タスクの実行
オペレータ	oper_role	サーバワイドのデータベースのバックアップとロード
Sybase サポート・センタ	sybase_ts_role	データベース構造の分析と修復
複写	replication_role	ユーザ・データの複写
分散トランザクション管理	dtm_tm_role	サーバ間のトランザクションのコーディネート
高可用性	ha_role	フェールオーバの管理と実行
モニタリングと診断	mon_role	パフォーマンスと診断のモニタリングの管理と実行
Job Scheduler 管理	js_admin_role	Job Scheduler の管理
Job Scheduler ユーザ	js_user_role、js_client_role	Job Scheduler によるジョブの作成と実行
リアルタイム・メッセージング	messaging_role	リアルタイム・メッセージングの管理と実行
Web サービス	webservices_role	Web Services の管理
キー管理者	keycustodian_role	暗号化キーの作成および管理

システム管理者の権限

システム管理者は以下のことを行います。

- アプリケーションに固有ではないタスクの処理
- Adaptive Server の任意アクセス制御システムの外部での作業

システム管理者の役割は、通常は特定の Adaptive Server ログインに付与されます。サーバ管理の作業量が 1 人で実行できる程度であれば、個人のログインではなく、Adaptive Server のインストール時に作成される “sa” アカウントを使用することもできます。インストール時に、“sa” アカウントのユーザは、システム管理者の役割、システム・セキュリティ担当者の役割、オペレータの役割を使用できることになります。“sa” アカウントのパスワードを知っていれば誰でも、そのアカウントにログインしてこれらの役割を持つことができます。

システム管理者が保護システムの外部で作業することは、安全対策の 1 つとなります。たとえば、データベース所有者が sysusers テーブル内のすべてのエントリを誤って削除してしまった場合でも、バックアップがあれば、システム管理者がそのテーブルをリストアできます。コマンドの中には、システム管理者しか発行できないものもあります。システム管理者しか発行できないコマンドは、disk init、disk refit、disk reinit、shutdown、kill、disk mirror、mount、unmount、および複数のモニタリングを行うコマンドです。

パーミッションを付与するとき、システム管理者はオブジェクト所有者として扱われます。システム管理者が、別のユーザのオブジェクトに対するパーミッションを付与すると、sysprotects と sp_helprotect の出力では、オブジェクト所有者の名前が付与者として表示されます。

システム管理者は、データベースにログインするときに、データベース所有者の ID を自動的に想定し、すべてのデータベース所有者の権限を使用します。この自動マッピングは、ユーザに割り当てられたエイリアスに関係なく実行されます。システム管理者は、dbcc コマンド、診断機能、データ・ページの読み取り、データやインデックスのリカバリなど、通常、データベース所有者用に予約されているタスクを実行できます。

システム・セキュリティ担当者の権限

システム・セキュリティ担当者は、Adaptive Server のセキュリティに関係する作業を実行します。これらの作業には、次のものがあります。

- システム・セキュリティ担当者、オペレータ、およびキー管理者の役割の付与
- 監査システムの管理
- パスワードの変更
- 新しいログインの追加
- ログインの削除

- ログイン・アカウントのロックとロック解除
- ユーザ定義の役割の作成と付与
- ネットワークベース・セキュリティの管理
- set proxy コマンドまたは set session authorization コマンドを使用するためのパーミッション付与

システム・セキュリティ担当者は、監査を有効にする必要があるため、すべてのデータベースにアクセスできますが、通常はデータベース・オブジェクトに対する特別なパーミッション (暗号化キーと暗号化カラムの decrypt パーミッションを除きます。『暗号化カラム・ユーザーズ・ガイド』を参照してください) は持ちません。sybsecurity データベースは例外で、このデータベースの sysaudits テーブルにはシステム・セキュリティ担当者以外はアクセスできません。システム・セキュリティ担当者しか実行できないシステム・プロシージャもあります。

システム・セキュリティ担当者は、ユーザの不注意による保護システムの変更を修復できます。たとえば、データベース所有者が自分のパスワードを忘れた場合、システム・セキュリティ担当者はパスワードを変更してデータベース所有者がログインできるようにします。

システム・セキュリティ担当者は、システム管理者とログインの管理責任を共有します。システム・セキュリティ担当者は、ログインの追加、ロック、およびロック解除を担当します。

システム・セキュリティ担当者は、ユーザ定義の役割を作成して、その役割をユーザ、他の役割、グループに付与することもできます。「ユーザに対する役割の作成と割り当て」(390 ページ) を参照してください。

オペレータの権限

オペレータの役割を付与されたユーザは、個々のデータベースの所有者にならなくても、サーバワイドでデータベースのバックアップとリストアを実行できます。オペレータの役割を付与されているユーザは、すべてのデータベースに対して次のコマンドを使用できます。

- dump database
- dump transaction
- load database
- load transaction
- checkpoint
- online database

システム・セキュリティ担当者はオペレータの役割を付与します。

Sybase サポート・センタ

Sybase 製品の保守契約を結んでいるサポート・センタの技術者は、サポート・センタの役割を使用して、トレース出力、一貫性チェック、データ構造へのパッチを通じて内部メモリ・データ構造とディスク上のデータ構造を表示できます。この役割は、問題の分析とデータのリカバリを手動で行うために使用されます。解決する問題によっては、データにアクセスするためにシステム標準の役割を追加する必要のある操作もあります。このような分析または修復を実行する場合、システム・セキュリティ担当者はこの役割を経験豊富な Sybase 技術者に対してのみ付与することをおすすめします。

複写の役割

Replication Server と ASE Replicator を管理するユーザには、複写の役割が必要です。この役割の詳細については、『Replication Server 管理ガイド』と『ASE Replicator ユーザーズ・ガイド』を参照してください。

分散トランザクション管理の役割

この役割は、分散トランザクション管理 (DTM) トランザクション・コーディネータが、システム・ストアド・プロシージャによるサーバ間のトランザクションの管理を可能にするために使用します。DTM XA インタフェースを使用するクライアントには、この役割が必要です。『Adaptive Server 分散トランザクション管理機能の使用』を参照してください。

高可用性の役割

高可用性の役割は、コマンドとストアド・プロシージャを通じてプライマリ・サーバとコンパニオン・サーバを管理する、高可用性サブシステムを設定するために必要です。『高可用性システムにおける Sybase フェールオーバの使用』を参照してください。

モニタリングと診断

この役割は、Adaptive Server のモニタリング・テーブルを管理するために必要です。モニタリング・テーブルのリモート・プロシージャ・コールの実行やモニタリングされたデータの収集の管理には、この役割が必要です。『パフォーマンス&チューニング・シリーズ：基本』を参照してください。

Job Scheduler の役割

Job Scheduler のオペレーションに対するパーミッションを管理するためのシステム標準の役割には、次の 3 つがあります。

- js_admin_role － Job Scheduler の管理に必要な役割であり、ストアド・プロシージャにアクセスして Job Scheduler の管理操作を修正、削除、実行できます。
- js_user_role － Job Scheduler のストアド・プロシージャを使用してスケジュール・ジョブを作成、修正、削除、実行するために必要な役割です。
- js_client_role － 定義済みジョブを使用できますが、ジョブを作成または変更することはできません。

詳細については、『Job Scheduler ユーザーズ・ガイド』を参照してください。

リアルタイム・メッセージングの役割

msgsend、msgrecv、および一部の sp_msgadmin コマンドを実行するために、リアルタイム・メッセージング・サブシステム (RTMS) で使用されます。詳細については、『Messaging Services ユーザーズ・ガイド』を参照してください。

Web Services の役割

この役割は、Web Services サブシステムで、create service、create existing service、drop service、および alter service コマンドを実行するために使用されます。『Web Services ユーザーズ・ガイド』を参照してください。

キー管理者の役割

キー管理者の役割は、暗号化キーの作成と変更、システム暗号化パスワードの設定、ユーザのキー・コピーの設定などのキー管理の責任があります。『暗号化カラム・ユーザーズ・ガイド』を参照してください。

ユーザ定義の役割

ユーザ定義の役割の計画

ユーザ定義の役割を実際に使用する前に、次のことを決定します。

- 作成する役割
- 各役割の責任
- 役割の階層における各役割の位置
- 階層内で相互排他的な関係にある役割と、その排他性をメンバシップ・レベルとアクティブ化レベルのどちらで設定するか

名前の重複を避けるには、命名規則に従ってユーザ定義の役割を作成するようにします。たとえば、役割名の末尾には “_role” を付けます。Adaptive Server は、そのような制限についてはチェックしません。

ユーザ定義の役割名は、ユーザ名と重複しないようにします。ある役割をユーザと同じ名前にする必要がある場合には、新しい役割を作成してそれに元の役割を組み込んでから、その新しい役割をユーザに付与することによって、矛盾を避けることができます。

作成する役割とその関係の計画が完了したら、ビジネス要件とユーザの責任に従って役割を割り付ける方法を決定してください。

ユーザがユーザ・セッションごとにアクティブ化できる役割の最大数は 127 です。

最小数の 15 には、Adaptive Server で用意されているシステム標準の役割が含まれます。

サーバワイドでアクティブ化できるユーザ定義の役割の最大数は 992 です。最初の 32 個の役割は、Sybase システム標準の役割用に予約されています。

ユーザ定義の役割の作成

役割の作成には、create role コマンドを使用します。構文は次のとおりです。

```
create role role_name [with passwd "password"
    [, {passwd expiration | min passwd length |
    max failed_logins } option_value ]]
```

各パラメータの意味は、次のとおりです。

- *role_name* − 新しい役割の名前です。
- *password* − オプションのパスワードです。この役割を使用しているユーザが指定する必要があります。
- *passwd expiration* − 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。

- *min passwd length* − 指定した役割に必要な最小のパスワード長を指定します。
- *max failed_logins* − 指定したログインに許可される、ログイン失敗の回数を指定します。
- *option_value* − *passwd expiration*、*min passwd length*、または *max failed_logins* の値を指定します。

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"
```

ユーザ定義役割を作成できるのは、システム・セキュリティ担当者だけです。

役割のパスワードの追加と削除

役割のパスワードを追加したり削除したりできるのは、システム・セキュリティ担当者だけです。

システム標準の役割またはユーザ定義の役割のパスワードを追加または削除するには、alter role コマンドを使用します。

```
alter role role_name
[add passwd password | drop passwd]
```

たとえば、oper_role にパスワード “oper8x” が必要となるようにするには、次のように入力します。

```
alter role oper_role add passwd oper8x
```

役割からパスワードを削除するには、次のように入力します。

```
alter role oper_role drop passwd
```

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。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。

役割の階層と相互排他性

この項では、役割の階層を設定し、役割の分割を実行する方法について説明します。

役割の相互排他性の定義と変更

2 つの役割間の相互排他性を定義するには、次の構文を使用します。

alter role *role1* { add | drop } exclusive { membership | activation } *role2*

たとえば、メンバシップ・レベルで、intern_role と specialist_role が相互排他となるように定義するには、次のように入力します。

```
alter role intern_role add exclusive membership
specialist_role
```

上記の例では、intern_role のメンバシップを持っているユーザが specialist_role のメンバにもならないように制限します。

sso_role と sa_role が、アクティブ化レベルで相互排他となるように定義するには、次のコマンドを入力します。このコマンドは、sso_role と sa_role のメンバであるユーザが、両方の役割を同時に持つことを禁止します。

```
alter role sso_role add exclusive activation sa_role
```

役割の階層の定義と変更

役割の階層を定義するには、初めに階層のタイプと役割を選択し、次に役割を別の役割に付与することによって階層を実装します。

次に例を示します。

```
grant role intern_role to specialist_role
grant role doctor_role to specialist_role
```

“specialist” に “doctor” と “intern” の両方が持つすべての権限を付与します。

役割 “super_user” に、システム標準の役割である sa_role と oper_role が含まれるような階層を作成するには、次のように指定します。

```
grant role sa_role to super_user
grant role oper_role to super_user
```

注意 パスワードの必要な役割が別の役割に含まれている場合、上位の役割が付与されているユーザは、下位の役割を使用するときもパスワードは必要ありません。上記の例では、役割 “doctor” に、通常はパスワードが必要であるとします。役割 “specialist” が付与されているユーザは “doctor” のパスワードを入力する必要はありません。“doctor” は “specialist” に含まれており、役割のパスワードは最高レベルの役割についてのみ要求されるためです。

役割の階層を作成するときは、次の規則に従います。

- ある役割を、それを直接含む別の役割に付与することはできません。これによって、重複が防止されます。

  上記の例では、役割 “doctor” を役割 “specialist” に付与することはできません。“specialist” には “doctor” が既に含まれているためです。

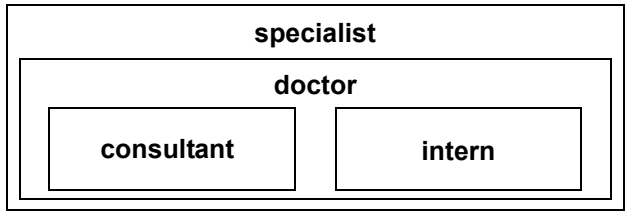
- ある役割を、それを直接含まない別の役割に付与することはできます。

  たとえば、図 14-2 では、役割 “specialist” に役割 “doctor” が既に含まれており、“doctor” に役割 “intern” が含まれていますが、“intern” を “specialist” に付与できます。その後で、“doctor” を “specialist” から削除しても、“specialist” に “intern” が含まれる状態は変わりません。

  図 14-2, では、“doctor” は役割 “consultant” のパーミッションを持っています。これは、“consultant” が “doctor” に付与されているためです。役割 “specialist” にも役割 “consultant” のパーミッションがあります。これは、“specialist” には役割 “doctor” が含まれ、役割 “doctor” には “consultant” が含まれるためです。

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

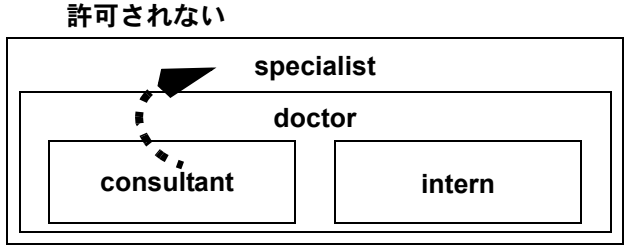
図 *14-2:* 明示的および暗黙的に付与された権限



- ある役割をその役割に含まれている別の役割に付与することはできません。これによって、階層内の「ループ」が回避されます。

  たとえば、図 14-3 では、役割 “specialist” を役割 “consultant” に付与することはできません。“consultant” は既に “specialist” に含まれているためです。

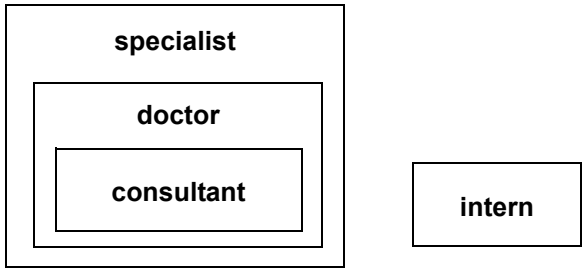
図 *14-3:* 付与者に含まれる役割に対する役割の付与



- システム・セキュリティ担当者がユーザに付与した役割に別の役割が含まれている場合は、そのユーザは、付与された役割に含まれるすべての役割におけるメンバシップを暗黙的に取得します。ただし、役割を直接アクティブ化または非アクティブ化できるのは、ユーザがその役割において明示的なメンバシップを持っている場合だけです。
- システム・セキュリティ担当者がある役割を別の役割に付与するとき、これらの役割がメンバシップ・レベルで明示的または暗黙的に相互排他である場合は、役割の付与はできません。

  たとえば、図 14-4 で、役割 “intern” が役割 “consultant” とメンバシップ・レベルで相互排他であると定義されている場合は、システム・セキュリティ担当者が “intern” を “doctor” に付与することはできません。

図 *14-4:* メンバシップでの相互排他性



- ユーザは直接付与された役割だけをアクティブ化、または非アクティブ化できます。

  たとえば、図 14-4 の階層で、役割 “specialist” がユーザに付与されたとします。ユーザには役割 “specialist” のすべてのパーミッションが付与されます。また、階層化されているので、役割 “doctor” と “consultant” のすべてのパーミッションも暗黙的に付与されます。ただし、このユーザがアクティブ化できるのは、役割 “specialist” だけです。“doctor” と “consultant” は、直接付与されたものではないので、アクティブ化はできません。「役割のアクティブ化と非アクティブ化」(401 ページ) を参照してください。

  役割を他の役割から取り消す方法は、役割を他の役割に付与する方法に似ています。これによって包含関係が削除されますが、包含関係は直接的なものでなければなりません。

次に例を示します。

- システム・セキュリティ担当者が役割 “specialist” から役割 “doctor” を取り消すと、“specialist” には役割 “consultant” も “intern” も含まれなくなります。
- 役割 “specialist” から役割 “intern” を取り消すことはできません。これは “intern” が “specialist” に直接含まれるものではないからです。

ログイン時のデフォルト・アクティブ化の設定

システム・セキュリティ担当者は、すべてのユーザについて、デフォルトの役割を変更できます。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。

ユーザが Adaptive Server にログインしたとき、デフォルトに設定された役割によっては、そのユーザの役割は必ずしもアクティブになりません。役割にパスワードが対応付けられている場合、ユーザは、set role コマンドを使用して、その役割をアクティブ化する必要があります。

システム・セキュリティ担当者またはユーザは、付与された役割をログイン時にデフォルトでアクティブ化するかどうかを決定します。sp_modifylogin は、各ユーザの個々の役割のデフォルト・ステータスを設定します。sp_modifylogin 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。

役割がログイン時にアクティブになるように設定するには、次の構文を使用します。

sp_modifylogin *loginname*, "add default role", *role_name*

複数のデフォルトの役割をユーザに割り当てるには、複数の sp_modifylogin コマンドを使用します。

役割がログイン時に非アクティブになるようにするには、次の構文を使用します。

sp_modifylogin *loginname*, "drop default role", *role_name*

たとえば、Ralph の intern_role がログイン時に自動的にアクティブになるようにデフォルト設定を変更するには、次のコマンドを実行します。

```
sp_modifylogin ralph, "add default role", intern_role
```

役割のアクティブ化と非アクティブ化

役割をアクティブ化しなければ、その役割の権限は得られません。デフォルトの役割はログイン時にアクティブにできません。パスワードがある役割は、ログイン時は必ず非アクティブになります。

役割をただちに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 [on|off]

パスワードが設定されている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]

たとえば、パスワード “sailing19” が設定されている役割 “financial_analyst” をアクティブ化するには、次のように入力します。

```
set role financial_analyst with passwd "sailing19" on
```

役割は必要なときにだけアクティブ化し、不要になったら非アクティブ化するようにしてください。たとえば、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ります。システム管理者の役割をオフにして、本来のユーザに戻るには、次のコマンドを使用します。

```
set role sa_role off
```

セッション中に付与された役割をすぐにアクティブ化するには、set role を使用してその役割をオンにしてください。

グループの設定とユーザの追加

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

グループ・レベルでは、パーミッションを付与および取り消すことができます。グループのパーミッションは、グループのメンバに自動的に渡されます。各データベースには、作成時にすべてのユーザが自動的に属する “public” という名前のグループが設定されています。sp_adduser を使用してユーザをグループに追加し、sp_changegroup を使用してユーザのグループを変更します。「ユーザのグループ・メンバシップの変更」(408 ページ) を参照してください。

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

ユーザ、グループ、ユーザ定義の役割の削除

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

ユーザの削除

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

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

```
sp_dropuser name_in_db
```

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

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

グループの削除

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

```
sp_dropgroup grpname
```

メンバを持っているグループを削除することはできません。削除しようとすると、そのグループのメンバの一覧を示すエラー・メッセージが表示されます。グループからユーザを削除するには、sp_changegroup を使用します。「ユーザのグループ・メンバシップの変更」(408 ページ) を参照してください。

ユーザ定義の役割の削除

役割を削除するには、システム・セキュリティ担当者は次のコマンドを使用します。*role_name* はユーザ定義の役割の名前です。

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

with override を指定すると、サーバワイドのすべてのデータベースで、その役割に付与されているアクセス権限がすべて取り消されます。

override オプションを使用しない場合は、すべてのデータベースでその役割に付与された権限をすべて取り消してから、役割を削除してください。この処理を行わないと、コマンドは失敗します。権限を取り消すには、revoke コマンドを使用します。

役割を削除する前に、メンバシップを削除する必要はありません。役割を削除すると、with override オプションを使用するかどうかにかかわらず、その役割内のユーザ・メンバシップは自動的に削除されます。

Adaptive Server ログイン・アカウントのロックおよび削除

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

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、sp_droplogin 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 sp_droplogin 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・アカウントのロックとロック解除

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

構文は次のとおりです。

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

各パラメータの意味は、次のとおりです。

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

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

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

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

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は、sp_droplogin を使用してユーザによる Adaptive Server へのアクセスを拒否できます。構文は次のとおりです。

```
sp_droplogin login_name
```

login_name で指定されるユーザがデータベース・ユーザまたはデータベース内のエイリアスとして存在する場合、sp_droplogin は失敗します。データベースからユーザを削除するには、sp_dropuser を使用してください。「ユーザの削除」(402 ページ) を参照してください。

スレッショルドを所有するログインのロック

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

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

ユーザ情報の変更

表 14-5 は、パスワード、デフォルト・データベース、デフォルト言語、フルネーム、グループの割り当ての変更に使用するシステム・プロシージャを示します。

表 *14-5:* ユーザ情報を変更するシステム・プロシージャ

作業	必要な役割	システム・プロシージャ	データベース
パスワードの変更	なし	sp_password	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	sp_password	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	sp_modifylogin	任意のデータベース

作業	必要な役割	システム・プロシージャ	データベース
自分のデフォルト・データベース、デフォルト言語、フルネームの変更	なし	sp_modifylogin	任意のデータベース
ログイン・アカウントのデフォルト・データベース、デフォルト言語、フルネームの変更	システム管理者またはシステム・セキュリティ担当者	sp_modifylogin	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

パスワードの変更

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

sp_ssladmin 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。

たとえば、パスワードを “3blindmice” から “2mediumhot” に変更するには、次のコマンドを使用します。

```
sp_password "3blindmice", "2mediumhot"
```

これらのパスワードは数字で始まっているので、引用符で囲まれています。

次の例では、“2tomato” というパスワードを持つシステム・セキュリティ担当者が、Victoria のパスワードを “sesame1” に変更します。

```
sp_password "2tomato", sesame1, victoria
```

新しいパスワードの要求

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

設定パラメータは、パスワード・ポリシー設定に置き換えられます。

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

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

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

null パスワード

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

```
sp_password null, "8M4LNCH"
```

注意　文の中で “null” を引用符で囲まないでください。

パスワードが失われた場合のログイン

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

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

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

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

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

```
dataserver -prsmith
```

sso_role のパスワードを再設定するには、次のように入力します。

```
dataserver -psso_role
```

ユーザ・デフォルトの変更

すべてのユーザは、sp_modifylogin を使用して自分のフルネーム、デフォルトのユーザ認証メソッド、デフォルト・データベース、デフォルト言語、およびデフォルトの役割を変更できます。sp_modifylogin を使用すると、パスワードの長さと有効期間を設定したり、ログイン試行の失敗回数を制限したり、ログイン時にログイン・スクリプトを自動的に実行するように指定したりできます。システム管理者は、すべてのユーザについてこれらの設定値を変更できます。構文は次のとおりです。

sp_modifylogin *login_name*, *option*, *value*

各パラメータの意味は、次のとおりです。

- *login_name* には、変更するアカウントのユーザの名前を指定します。
- *option* には、変更するオプションを指定します。使用できるオプションのリストについては、『リファレンス・マニュアル：プロシージャ』の「sp_modifylogin」を参照してください。
- *value* は、指定するオプションの新しい値です。

sp_modifylogin 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、sp_modifylogin 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” のデフォルト・データベースを pubs2 に変更します。

```
sp_modifylogin anna, defdb, 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。

```
sp_modifylogin claire, deflanguage, french
```

次の例では、“mtwain” のフルネームを “Samuel Clemens” に変更します。

```
sp_modifylogin mtwain, fullname, "Samuel Clemens"
```

ユーザのグループ・メンバシップの変更

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

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

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

sp_changegroup の構文は次のとおりです。

```
sp_changegroup grpname, username
```

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

```
sp_changegroup management, jim
```

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

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

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

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

ユーザの所属グループの割り当てはいつでも変更できます。

ユーザ・プロセス情報の変更

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

以下は、set コマンドの構文の一部です。

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

client_name はクライアントに割り当てる名前、*host_name* はクライアントの接続元ホストの名前、*application_name* は Adaptive Server に接続しているアプリケーションです。これらのパラメータは、sysprocesses テーブルのカラム clientname、clienthostname、clientapplname に格納されます。

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

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

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

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

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

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

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

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

データベース内でのエイリアスの使用

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

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

注意 1 つのデータベース内で複数のユーザが同じエイリアスを使用できますが、その場合も、各ユーザが実行するデータベース操作を監査することによって、個々の責任を明確にすることが可能です。「第 18 章 監査」を参照してください。

エイリアスを使用して得られる集合ユーザ ID は、データベース・オブジェクトの集合所有権を意味します。たとえば、ユーザ “loginA” がデータベース db1 の dbo in にエイリアスとして指定されている場合は、db1 の “loginA” で作成されたすべてのオブジェクトが dbo によって所有されます。ただし、Adaptive Server はログイン名と作成者のデータベース・ユーザ ID については、オブジェクトの所有権を具体的に記録します。「具体的 ID」(543 ページ) を参照してください。そのデータベース内でオブジェクトを具体的に所有している場合は、データベースからエイリアスを削除することはできません。

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

エイリアスの追加

ユーザのエイリアスを追加するには、sp_addalias を使用します。

```
sp_addalias loginame, name_in_db
```

各パラメータの意味は、次のとおりです。

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

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

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

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

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

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

エイリアスの削除

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

sp_dropalias *loginame*

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

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

エイリアス情報を取得する方法

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

```
sp_helpuser dbo

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

(1 row affected)

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

ユーザ情報を取得する方法

表 14-6 は、ユーザ、グループ、現在の Adaptive Server の使用状況に関する情報を表示するために使用するプロシージャを示します。

表 *14-6: Adaptive Server* のユーザとグループの情報の表示

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

ユーザとプロセスをレポートする方法

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

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

各パラメータの意味は、次のとおりです。

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

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

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

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

```
spid    status  loginame   origname  hostname  blk  dbname              cmd
----    -------  --------  ---------  --------  ---  ------    -------------
   1   running        sa        sa    sunbird    0     pubs2           SELECT
   2  sleeping      NULL      NULL               0    master  NETWORK HANDLER
   3  sleeping      NULL      NULL               0    master   MIRROR HANDLER
   4  sleeping      NULL      NULL               0    master    AUDIT PROCESS
   5  sleeping      NULL      NULL               0    master  CHECKPOINT SLEEP

(5 rows affected, return status = 0)
```

sp_who の出力では、システム・プロセスの *loginame* はすべて NULL です。

ログイン・アカウントに関する情報の取得

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

```
sp_displaylogin [loginame | wildcard]
```

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

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

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

```
sp_displaylogin 'sa'

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

データベース・ユーザ情報を取得する方法

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

sp_helpuser [*name_in_db*]

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

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

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

ユーザの名前と ID を表示する方法

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

表 *14-7: suser_id システム関数と suser_name システム関数*

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

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

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

```
select suser_id("sandy")
------
     3
```

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

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

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

表 *14-8: user_id* システム関数と *user_name* システム関数

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

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

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

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

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

役割に関する情報の表示

表 14-9 は、役割に関する情報の表示に使用するシステム・プロシージャと関数を示します。

表 *14-9:* 役割について参照する情報

表示する情報	使用	参照箇所
役割名の役割 ID	role_id システム関数	「役割 ID と役割名の表示」(417 ページ)
役割 ID の役割名	role_name システム関数	「役割 ID と役割名の表示」(417 ページ)
システム標準の役割	show_role システム関数	「アクティブなシステム標準の役割の表示」(417 ページ)
役割階層、およびユーザに付与された役割	sp_displayroles システム・プロシージャ	「役割の階層の表示」(418 ページ)
役割階層内で、ある役割に他の役割が含まれているかどうか	role_contain システム関数	「階層内のユーザ定義の役割の表示」(418 ページ)
2 つの役割が相互排他的かどうか	mut_excl_roles システム関数	「相互排他性の判別」(418 ページ)
現在のセッションに対してアクティブな役割	sp_activeroles システム・プロシージャ	「役割のアクティブ化の判別」(418 ページ)
プロシージャを実行するために正しい役割がアクティブ化されているかどうか	proc_role システム関数	「ストアド・プロシージャ内の役割の検査」(418 ページ)

表示する情報	使用	参照箇所
ログイン (付与された役割を含む)	sp_displaylogin システム・プロシージャ	「ログイン・アカウントに関する情報の取得」(414 ページ)
ユーザ、グループ、または役割についてのパーミッション	sp_helprotect システム・プロシージャ	「パーミッションを表示する方法」(565 ページ)

役割 ID と役割名の表示

役割の名前がわかっている場合に、その役割 ID を表示するには、次の構文を使用します。

role_id(*role_name*)

すべてのユーザが role_id を実行できます。役割が有効ならば、role_id は、サーバワイドでのその役割の ID (srid) を返します。syssrvroles システム・テーブルの srid カラムに役割 ID が格納され、name カラムに役割名が格納されています。役割が無効な場合、role_id は NULL を返します。

役割 ID がわかっている場合に、その役割名を表示するには、role_name を使用します。

role_name(*role_id*)

すべてのユーザが role_name を実行できます。

アクティブなシステム標準の役割の表示

指定したログインの現在アクティブなシステム標準の役割を表示するには、show_role を使用します。

show_role()

ログインに対してシステム標準の役割が 1 つもアクティブ化されていない場合、show_role は NULL を返します。実行するユーザがデータベース所有者であり、別のユーザになり代わるために setuser を実行した後で show_role を実行した場合は、show_role はなり代わる別のユーザのアクティブなシステム標準の役割ではなく、そのユーザ自身のシステム標準の役割を返します。

すべてのユーザがシステム関数 show_role を実行できます。

注意　システム関数 show_role を実行しても、ユーザ定義の役割についての情報は表示されません。

役割の階層の表示

sp_displayroles を使用すると、ログイン名に付与されたすべての役割を表示することや、役割の階層ツリー全体をテーブル形式で表示することができます。

```
sp_displayroles {login_name | rolename [, expand_up | expand_down]}
```

すべてのユーザが、sp_displayroles を実行して各自の役割を表示できます。他のユーザに付与された役割に関する情報を表示できるのは、システム・セキュリティ担当者だけです。

階層内のユーザ定義の役割の表示

指定した役割に、指定した別の役割が含まれているかどうかを調べるには、role_contain を使用します。

```
role_contain (["role1", "role2"])
```

role1 が *role2* に含まれている場合、role_contain は 1 を返します。

すべてのユーザが role_contain 関数を実行できます。

相互排他性の判別

ユーザに割り当てられた 2 つの役割が相互排他の関係にあるかどうかと、その役割がどのレベルで相互排他であるかを調べるには、mut_excl_roles を使用します。

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

すべてのユーザが mut_excl_roles 関数を実行できます。指定した役割、または指定した役割に含まれる役割が相互排他の関係にある場合、mut_excl_roles は 1 を返します。役割が相互排他の関係にない場合、mut_excl_roles は 0 を返します。

役割のアクティブ化の判別

Adaptive Server の現在のログイン・セッションでアクティブな役割をすべて表示するには、次のコマンドを使用します。

```
sp_activeroles [expand_down]
```

expand_down を指定すると、ユーザに付与された役割に含まれるすべての役割の階層が表示されます。

すべてのユーザが sp_activeroles を実行できます。

ストアド・プロシージャ内の役割の検査

特定の役割を持つユーザだけがストアド・プロシージャを実行できることを保証するには、そのプロシージャの中で proc_role を使用します。特定のストアド・プロシージャに対する不正なアクセスを防止して安全を保証するメカニズムは、proc_role だけです。

grant execute を使用すると、ストアド・プロシージャに対する実行パーミッションを、指定の役割が付与されているすべてのユーザに付与できます。同様に、revoke execute を使用すると、このパーミッションを削除できます。

ただし、grant execute では、指定の役割を持たないユーザにストアド・プロシージャの実行パーミッションが付与されることを防ぐことはできません。たとえば、システム管理者以外のユーザに、ストアド・プロシージャを実行するパーミッションが決して付与されないようにするには、そのストアド・プロシージャの中で proc_role を使用し、呼び出しを行うユーザに正しい役割があるかどうかを検査します。

proc_role は、必要な役割の文字列を受け取り、呼び出し元がその役割を所有していれば 1 を返します。所有していなければ、0 を返します。

たとえば、次のプロシージャは、proc_role を使用して、ユーザが役割 sa_role を持つかどうかを確認しています。

```
create proc test_proc
as
if (proc_role("sa_role") = 0)
begin
    print "You don't have the right role"
    return -1
end
else
    print "You have System Administrator role"
    return 0
```

パスワードとログイン・ポリシーの設定

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

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

- 無効なパスワードが何回入力されたらログインや役割を自動的にロックするか、その回数を指定する
- パスワードが失われた場合のログイン
- 手動によるログインと役割のロックとロック解除
- ログイン・パスワード情報の表示
- サーバワイドまたは特定のログインや役割に対して、最小パスワード長 (minimum password length) を指定する
- 複雑なログインのパスワードのチェック
- ログインのカスタムのパスワード・チェックの有効化

- パスワード有効期間の設定
- 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
- ディスクへのパスワードの保管に SHA-256 アルゴリズムのみを使用する
- ログインのパスワード文字セットを考慮する
- アップグレードとダウングレードの動作の実行
- 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

ログインを試行できる最大回数の設定と変更

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

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

❖ サーバワイドでの *maximum failed logins* の設定

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

  許可されるログイン失敗回数を設定するには、次のように入力します。

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

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

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

- sp_addlogin を使用してログインを作成するときに、そのログインに対する maximum failed logins を設定できます。

  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 2 に設定します。

  ```
  sp_addlogin joe, "Djdiek3", pubs2, null, null, null, null, 2
  ```

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

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

- create role を使用して役割を作成するときに、その役割に対する maximum failed logins を設定できます。

  この例では、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 に対する maximum failed logins を 20 に設定します。

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

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

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

- 既存のログインに対する maximum failed logins を設定または変更するには、sp_modifylogin を使用します。

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

  ```
  sp_modifylogin "joe", "max failed_logins", "40"
  ```

  注意　*value* パラメータのデータ型は character です。したがって、数値には引用符が必要です。

  sp_modifylogin 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。構文と規則の詳細については、「sp_modifylogin」を参照してください。

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

❖ 特定の役割の *maximum failed logins* の変更

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

  例 1 physician_role に対する maximum failed logins を 5 に変更します。

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

  例 2 すべての役割に対する maximum failed logins を無効にする設定を削除します。

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

  maximum failed logins を使用するための構文と規則の詳細については、「alter role」を参照してください。

パスワードが失われた場合のログイン

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

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

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

たとえば、次のように入力してサーバを起動したとします。

```
dataserver -psa_role
```

Adaptive Server は次のメッセージを表示します。

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

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

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

ログインと役割のロックとロック解除

次のような場合にログインまたは役割をロックできます。

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

❖ ログインのロックとロック解除

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

  次に例を示します。

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

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

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

❖ 役割のロックとロック解除

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

  次に例を示します。

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

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

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

❖ サーバの起動時のログインと役割のロック解除

- 自動ログイン・ロックアウトを使用すると、サイトではログインをロック解除できるすべてのアカウント (システム管理者とシステム・セキュリティ担当者) がロックされた状態になることがあります。Adaptive Server の起動時に特定のログインまたは役割のロックを解除するには、dataserver ユーティリティの -u フラグを使用します。

  『ユーティリティ・ガイド』の「dataserver」を参照してください。

パスワード情報の表示

この項では、ログインと役割のパスワード情報の表示方法について説明します。

❖ **特定のログインのパスワード情報の表示**

- ログインのパスワード設定を表示するには、sp_displaylogin を使用します。

  この例では、ログイン joe についての情報が表示されます。

```
sp_displaylogin joe

Suid: 3
Loginame:joe
Fullname:
Default Database:master
Default Language:
Auto Login Script:
Configured Authorization:
Locked:NO
Date of Last Password Change:Sep 22 2008  3:50PM
Password expiration interval: 0
Password expired:NO
Minimum password length: 6
Maximum failed logins: 1
Current failed login attempts: 2
Authenticate with:ANY
Login Password Encryption:SHA-256
Last login date:Sep 18 2008 10:48PM
```

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

❖ **特定の役割のパスワード情報の表示**

- 役割のパスワード設定を表示するには、sp_displayroles を使用します。

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

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

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

パスワードが 1 文字以上あるかどうかの検査

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

次の例では、パスワードの検査機能をアクティブにします。

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

この例では、パスワードの検査機能を非アクティブにします。

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

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

minimum password length の設定と変更

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

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

システム・セキュリティ担当者は、以下のものを指定できます。

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

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

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

- sp_addlogin を使用してログインを作成するときに、そのログインに対する minimum password length を設定できます。

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

  ```
  sp_addlogin joe, "Djdiek3", @minpwdlen=8
  ```

  minimum password length 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：プロシージャ』の「sp_addlogin」を参照してください。

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

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

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

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

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

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

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

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

  例 1 ログイン “joe” の minimum password length を 8 文字に変更します。

  ```
  sp_modifylogin "joe", @option="min passwd length",
  @value="8"
  ```

  注意 *value* パラメータのデータ型は character です。したがって、数値には引用符が必要です。

  例 2 すべてのログインに対する minimum password length のオーバライドを 8 文字に変更します。

  ```
  sp_modifylogin "all overrides", @option="min passwd
  length", @value="8"
  ```

  例 3 すべてのログインに対する minimum password length のオーバライドを削除します。

  ```
  sp_modifylogin "all overrides", "min passwd length",
  @value="-2"
  ```

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

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

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

  例 1 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
  ```

  例 2 すべての役割の minimum password length を無効にします。

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

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

複雑なパスワード・チェック

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

個々のオプションをオフにするには、次のように入力します。

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

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

```
sp_passwordpolicy 'clear'
```

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

単純なパスワードの禁止

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

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

このオプションを設定するには、次のように入力します。

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

単純なパスワードを禁止する場合は、ログイン名をパスワードの部分文字列として使用できません。パスワードは複雑な文字列に設定する必要があります。次に例を示します。

```
sp_password 'old_complex_password', BHotAcha789, johnd
```

ログイン johnd のパスワードは、BHotAcha789 で、ログイン名は部分文字列として含まれていません。

ただし、ログイン・パスワードを次のように変更すると、ログイン johnd が新しいパスワード johnd123 の部分文字列となり、コマンドは失敗します。

```
sp_password 'old_complex_password', johnd123, johnd
```

カスタムの複雑なパスワード・チェック

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

これらのストアド・プロシージャは、master データベースで定義および配置されており、Adaptive Server による複雑なパスワード・チェック中に自動的に呼び出され、この時点でログインがそれぞれ破棄されます。これらのカスタム・ストアド・プロシージャの作成例については、「カスタムのパスワード・チェックの有効化」(433 ページ) を参照してください。

パスワードの文字数の指定

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

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

- min lower char in password － パスワードの小文字の最小文字数。有効な値は次のとおりです。
  - 0 ～ 16 － パスワードに必要な大文字の文字数。
  - -1 － パスワードに大文字を含めることはできない。
- minimum password length － 最小パスワード長。最小パスワード長は 0 ～ 30 の範囲で設定できます。指定する値は、他の最小要件をすべて組み合わせた長さ以上にする必要があります。たとえば、次のように設定している場合は、minimum password length を 10 以上に設定する必要があります。
  - minimum digits in password を 3 に設定
  - minimum special characters in password を 2 に設定
  - minimum uppercase characters in password を 2 に設定
  - minimum lowercase characters in password を 3 に設定
- password expiration － 期限が切れるまでの、パスワードが存在できる日数。この値はグローバルな単位で指定します。デフォルトでは無効です。有効な値は次のとおりです。
  - 0 － パスワードの期限は切れない。
  - 1 ～ 32767 － 期限が切れるまでの、パスワードが存在できる日数。
- password exp warn interval － パスワードの期限が切れるまで、パスワード有効期限の警告メッセージを表示する間隔 (日数)。これらのメッセージは、パスワードが変更されるか、期限が切れるまで、成功したすべてのログインで表示されます。この値は、パスワード有効期限以下の値にする必要があります。デフォルトでは無効です。

  有効値は 0 ～ 365 です。
- maximum failed logins － ログインがロックされるまで実行できる、ログイン失敗の最大回数。この値はグローバルに指定します。デフォルトでは無効です。有効な値は次のとおりです。
  - 0 － ログイン失敗回数に関係なく、ログインはロックされない。
  - 1 ～ 32767 － ログインがロックされるまでに許可されるログイン失敗回数。

- expire login では、システム・セキュリティ担当者がログインを作成またはリセットすると、ログインのステータスを期限切れに変更します。ログインは、初回ログイン時にパスワードを変更する必要があります。デフォルトでは無効です。有効な値は次のとおりです。
  - 0 － 新しいログインまたはリセットされたログインには期限を設定しない。
  - 1 － 新しいログインまたはリセットされたログインの期限が切れた場合は、初回ログイン時にパスワードをリセットする必要がある。

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

複雑なパスワード・オプションの相互チェック

複雑なパスワード・オプションには、次の対話的な機能を持つものがあります。

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

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

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

複雑なパスワード・チェックの設定

表 *14-10:* 複雑なパスワード・チェック

Adaptive Server 認証のパスワード･チェックとポリシー	sp_configure を使用して指定される設定パラメータ	sp_passwordpolicy を使用して指定される複雑なパスワード・オプション	sp_modifylogin 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_logins
単純なパスワードの禁止	該当なし	disallow simple passwords	該当なし
パスワードの特殊文字数	該当なし	min special char in password	該当なし
パスワードの大文字数	該当なし	min upper char in password	該当なし
パスワードの小文字数	該当なし	min lower char in password	該当なし
パスワード有効期限の警告間隔	該当なし	password exp warn interval	該当なし
初回ログイン時のパスワードのリセット	該当なし	expire login	該当なし
カスタムの複雑なパスワード・チェック	該当なし	該当なし	該当なし

複雑なパスワード・オプションは次のレベルで設定できます。

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

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

パスワード・オプションを適用すると、優先順位は次のようになります。

1　既存のログイン単位のパラメータ

2　複雑なパスワード・オプション

3　既存のグローバル・パスワード・オプション

例

例 1　次のように入力すると、“johnd” の最小パスワード長は 6 に設定されます。

```
sp_addlogin @login_name = 'johnd',
      @passwd = 'complex_password',
      @minpwdlen = 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作成します。

```
sp_password @caller_password = 'old_complex_password',
@new_password = 'abcd123', @login_name = 'johnd'
```

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

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

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

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

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

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

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

```
sp_password @caller_password = 'old_complex_password',
@new_password = 'abcd', @login_name = 'johnd'
```

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

例 3 次の例は、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。

```
sp_addlogin @login_name = 'johnd', @passwd =
'complex_password', @minpwdlen = 4
```

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

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

次に、ログイン johnd のパスワードを次のように作成します。

```
sp_password @caller_password = 'old_complex_password',
@ new_password = 'abcde', @login_name = 'johnd'
```

Adaptive Server では、次の順序でチェックを実行します。

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

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

Adaptive Server は新しいパスワードを作成せずに、エラー・メッセージを示します。

新しいパスワードを作成するには、すべてのチェックに合格する必要があります。

カスタムのパスワード・チェックの有効化

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

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

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

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

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

```
as

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


select @changedby = suser_name()

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

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

if not exists   ( select 1 from master..pwdhistory
                where name = @loginame
                and   password = @encrypted_pwd )
begin
      insert master..pwdhistory
      select @loginame, internal_encrypt(@new_password),
             @current_time, @changedby
      return(0)
end
else
begin
      raiserror 22001  --user defined error message
end
end
```

sp_addmessage を使用して、ユーザ定義のメッセージ 22001 を追加します。raiserror 22001 は、カスタムの複雑なパスワード・チェックのエラーが発生し、それが原因で sp_addlogin または sp_password が失敗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 つのパラメータが定義され、master データベースにインストールされると、これらのパラメータは複雑なパスワード・チェック中に動的に呼び出されます。

パスワードのログインと役割の有効期間の設定

システム管理者とシステム・セキュリティ担当者は次のことができます。

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

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

- ログイン・アカウントごと、または役割ごとに割り当てたパスワード有効期間は、システム全体にわたるパスワード有効期間の値よりも優先される。これによって、システム・セキュリティ担当者のパスワードなどの機密性の高いアカウントまたは役割には比較的短いパスワード有効期間を指定し、匿名ログインなどの機密性の低いアカウントには比較的長い有効期間を指定できる。
- パスワードの有効期間が切れているログインまたは役割は、直接アクティブにはならない。
- パスワードは、password expiration interval によって指定された日数が過ぎた後、パスワードを最後に変更した日に有効期限が切れます。

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

12.x より前のパスワードにはパスワード有効期間が無効

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

パスワードによる保護の迂回

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

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

次に例を示します。

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

- financial_assistant

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

- accounts_officer

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

- chief_financial_officer

  ```
  create role chief_financial_officer
  ```

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

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

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

```
grant role chief_financial_officer to bob
```

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

```
set role chief_financial_officer on
```

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

新規ログインのパスワードの有効期間の作成

新規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、sp_addlogin を使用します。

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

```
sp_addlogin joe, "Djdiek3", null, null, null, 2
```

“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 2 日後、またはパスワードを最後に変更した日から 2 日後に有効期限が切れます。

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

新規役割のパスワードの有効期間の作成

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

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

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

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

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

パスワードの作成日の刻印

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

ログインや役割に設定されているパスワード有効期間の変更または削除

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

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

```
sp_modifylogin "joe", @option="passwd expiration", @value="5"
```

パスワードは、パスワード有効期限を過ぎた日から 5 日後に期限切れになります。

注意 *value* パラメータのデータ型は character です。したがって、数値には引用符が必要です。

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

ネットワーク上でのログイン・パスワードの保護

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

Adaptive Server がクライアントにログイン・プロトコルの使用を要求するように設定できます。Adaptive Server の設定パラメータ net password encryption reqd を設定して、ユーザ名とパスワードに基づくすべての認証で RSA 非対称暗号化の使用を要求できる。「net password encryption required」(173 ページ) を参照してください。

非対称キー・ペアの生成

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

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

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

次のコマンドを使用すると、いつでもキー・ペアを生成できます。

```
sp_passwordpolicy "regenerate keypair"
```

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

キー・ペアの生成時刻を指定するには、次のコマンドを使用します。

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

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。

サーバ・オプション “net password encryption”

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

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

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

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

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

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

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

旧バージョンとの互換性

- Sybase ではネットワーク上でパスワードを保護するために RSA アルゴリズムを使用するようおすすめしています。
- RSA アルゴリズムを使用するには、Adaptive Server バージョン 15.0.2 と新しい Connectivity SDK クライアント (バージョン 15.0 ESD#7 以降) が必要です。Sybase では net password encryption reqd 設定パラメータと net password encryption サーバ・オプションを用意することによって 15.0.2 より古いバージョンでの設定と同じ設定を使用できるようにして、旧バージョンのクライアントやサーバとの互換性を維持しています。

- RSA アルゴリズムをサポートしない古いクライアントではそのプロパティに、バージョン 12.0 以前から使用されてきた Sybase 独自のアルゴリズムによるパスワード暗号化を設定できます。そうすれば、Adaptive Server は Sybase 独自のアルゴリズムを使用します。
- RSA アルゴリズムも Sybase 独自のアルゴリズムもサポートする新しいクライアントでは、両方のアルゴリズムをプロパティに設定します。そのようなクライアントと通信するとき Adaptive Server 15.0.2 は RSA 暗号化を使用します。15.0.2 より古い Adaptive Server は Sybase 独自のアルゴリズムを使用します。

ディスクとメモリに保管されているログイン・パスワードの保護

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

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

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

パスワードが新旧両形式で保管されている限り、Adaptive Server を、ユーザ・パスワードをリセットせずに Adaptive Server 15.0 または 15.0.1 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。「旧バージョンとの互換性」(439 ページ) を参照してください。

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。

SHA-256 アルゴリズムのみの使用

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
2> go
Suid: 70
Loginame: login933
```

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

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

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

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

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

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

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

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

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

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

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

例 2

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

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

Old password encryption algorithm found for login name login1000, suid 3,
ver1 =5, ver2 = 0, resetting password to EcJxKmMvOrDsC4
Old password encryption algorithm found for login name login999, suid 4,
ver1 =5, ver2 = 0, resetting password to MdZcUaFpXkFtM1
Old password encryption algorithm found for login name login998, suid 5,
ver1 =5, ver2 = 0, resetting password to ZePiZdSeMqBdE6
Old password encryption algorithm found for login name login997, suid 6,
ver1 =5, ver2 = 0, resetting password to IfWpXvGlBgDgW7
Old password encryption algorithm found for login name login996, suid 7,
ver1 =5, ver2 = 0, resetting password to JhDjYnGcXwObI8
Old password encryption algorithm found for login name login995, suid 8,
ver1 =5, ver2 = 0, resetting password to QaXlRuJlCrFaE6
Old password encryption algorithm found for login name login994, suid 9,
ver1 =5, ver2 = 0, resetting password to HlHcZdRrYcKyB2
Old password encryption algorithm found for login name login993, suid 10,
ver1 =5, ver2 = 0, resetting password to UvMrXoVqKmZvU6
Old password encryption algorithm found for login name login992, suid 11,
ver1 =5, ver2 = 0, resetting password to IxIwZqHxEePbX5
Old password encryption algorithm found for login name login991, suid 12,
ver1 =5, ver2 = 0, resetting password to HxYrPyQbLzPmJ3
Old password encryption algorithm usage eliminated from 10 login accounts,
changes are committed.
(return status = 1)
```

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

パスワード文字セットの考慮事項

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

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

sp_password *old_passwd*, *new_passwd*

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

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

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

1 クリア・テキスト・パスワード文字列を UTF-16 に変換。

2 UTF-16 文字列をネットワーク・バイト順序に変換。

3 ランダム・バイトの小さなバッファをソルトとしてパスワードの末尾に付加。

4 SHA-256 ハッシュ・アルゴリズムを適用。

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

認証過程は次のようになります。

1 クリア・テキスト・パスワード文字列を UTF-16 に変換。

2 UTF-16 文字列をネットワーク・バイト順序に変換。

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

4 ハッシュ・アルゴリズムを適用。

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

アップグレードとダウングレードの動作

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

アップグレードされた *master* データベースの動作の変化

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

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

アップグレード後の最初のログイン認証では

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

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

新しい *master* データベースの動作の変化

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

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

```
sp_passwordpolicy "list", "allow password downgrade"
```

アップグレードしてからダウングレードした後のパスワード暗号化の保持

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

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

アップグレード前の領域の追加

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

暗号化アルゴリズムとパスワード・ポリシー

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

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

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

ダウングレード

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

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

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

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

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

Checking databases for downgrade readiness.

There are no errors which involve encrypted columns.

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

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

Password will be reset during the downgrade phase.

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

sp_droplogin 'probe'
```

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

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

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

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

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

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

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

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

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

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

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

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

Total number of passwords reset during downgrade = 1

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

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

```
00:00000:00006:2007/05/21 05:34:07.81 server  Preparing ASE downgrade from 1502 to 1501.
00:00000:00006:2007/05/21 05:35:59.09 server  Preparing ASE downgrade from 1502 to 1501.
00:00000:00006:2007/05/21 05:35:59.19 server  Starting downgrading ASE.
00:00000:00006:2007/05/21 05:35:590.20 server  Downgrade :Downgrading login passwords.
00:00000:00006:2007/05/21 05:35:590.22 server  Downgrade :Starting password downgrade.
00:00000:00006:2007/05/21 05:35:590.23 server  Downgrade :Removed sysattributes rows.
00:00000:00006:2007/05/21 05:35:590.23 server  Downgrade :Updated 1 passwords.
00:00000:00006:2007/05/21 05:35:590.24 server  Downgrade :Removed columns in syslogins -
lastlogindate, crdate, locksuid, lockreason, lockdate are removed.
00:00000:00006:2007/05/21 05:35:590.26 server  Downgrade :Truncated password lengths.
00:00000:00006:2007/05/21 05:35:590.28 server  Downgrade :Successfully completed password
downgrade.
00:00000:00006:2007/05/21 05:35:590.28 server  Downgrade :Marking stored procedures to
be recreated from text.
00:00000:00006:2007/05/21 05:36:030.69 server  Downgrade :Dropping Sysoptions system
table.
00:00000:00006:2007/05/21 05:36:030.81 server  Downgrade :Setting master database minor
upgrade version.
00:00000:00006:2007/05/21 05:36:03.83 server  Downgrade :Setting user databases minor
upgrade version.
00:00000:00006:2007/05/21 05:36:03.90 server  ASE downgrade completed.
```

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

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

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

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

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

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

役割パスワードが無効になるように設定するには、次のコマンドを使用します。

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

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

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

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

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

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

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

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

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

```
sp_passwordpolicy list, "allow password downgrade"
```

結果セットには現在の値とその意味を説明するメッセージが含まれています。

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

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

value    message
```

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

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

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

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

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

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

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

最後のログインと非アクティブ・アカウントのロック

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

- 作成日を追跡する。
- アカウントに最後にログインした日時を記録する。
- 非アクティブになっているためロックしてもよいアカウントを特定する。
- アカウントがロックされた理由とアカウントをロックしたユーザ ID を記録する。

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

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

ログイン作成時に、crdate カラムはそのときの日時に設定されます。

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

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

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

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

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

```
sp_locklogin '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、および lockreason の各カラムの設定は Adaptive Server 内部で処理されます。表 14-11 は locksuid の lockreason と値を示します。

表 *14-11: locksuid* の値と理由

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

高可用性環境でのパスワードの使用

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

高可用性の設定

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

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

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

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

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

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

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

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

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

アップグレード後に更新されたパスワード

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

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

ライセンス使用状況のモニタリング

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

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

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

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

ライセンスがカウントされる仕組み

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

License Use Monitor の設定

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

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

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

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

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

ハウスキーピング・タスクを使用したライセンス使用状況のモニタリング

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

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

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

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

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

ユーザ・ライセンス数のロギング

Adaptive Server をインストールまたはアップグレードするときに、master データベース内に syblicenseslog システム・テーブルが作成されます。表 14-12 に示すように、License Use Monitor は 24 時間ごとに、syblicenseslog 内のカラムを更新します。

表 *14-12: syblicenseslog* テーブル内のカラム

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

次は、syblicenseslog の例です。

```
status logdate                    maxlicenses
------ -------------------------- -----------
     0    Jul 17 1998 11:43AM           123
     0    Jul 18 1998 10:47:00AM        147
     1    Jul 19 1998 10:51:00AM        154
     0    Jul 20 1998 10:55:00AM        142
     0    Jul 21 1998 10:58:00AM        138
     0    Jul 21 1998  3:14PM           133
```

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

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

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

使用状況に関する情報の表示 : チャージバック・アカウンティング

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

現在使用量の統計のレポート

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

現在のアカウンティング合計の表示

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

新しいアカウンティング期間の開始

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

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

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

アカウンティング統計を追加する間隔の指定

システム管理者は、設定パラメータを使用して、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累積クロック・チック数を指定するには、cpu accounting flush interval 設定パラメータを使用します。デフォルト値は 200 です。次に例を示します。

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

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

```
select @@timeticks
```

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

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

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

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

第 15 章

リモート・サーバの管理

この章では、各 Adaptive Server のシステム管理者とシステム・セキュリティ担当者が、「リモート・プロシージャ・コール」(RPC) を使用するために実行する必要がある手順について説明します。

概要

ローカル Adaptive Server 上のユーザは、リモート Adaptive Server 上のストアド・プロシージャを実行できます。RPC を実行すると、リモート・プロセスの結果が、呼び出し元プロセスに送信されます。これは通常、ユーザの画面に表示されます。

RPC を使用できるようにするには、各 Adaptive Server のシステム管理者とシステム・セキュリティ担当者が次の手順を実行する必要があります。

- ローカル・サーバ上
  - システム・セキュリティ担当者が sp_addserver を使用して、システム・テーブル master..sysservers にローカル・サーバとリモート・サーバを登録します。
  - リモート・サーバを、ローカル・サーバの *interfaces* ファイルまたはディレクトリ・サービスに登録します。
  - ローカル・サーバを再起動します。これによって、グローバル変数 *@@servername* がローカル・サーバの名前に設定されます。この変数が正しく設定されていない場合は、ローカル・サーバからリモート・サーバ上で RPC を実行することはできません。

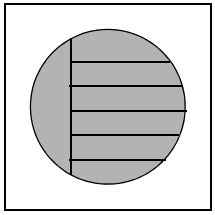
- リモート・サーバ上
  - システム・セキュリティ担当者が sp_addserver を使用して、システム・テーブル master..sysservers に RPC の発信元のサーバを登録します。
  - リモート・プロシージャを発信するユーザがこのサーバにアクセスできるようにするために、システム・セキュリティ担当者が sp_addlogin を実行し、システム管理者が sp_addremotelogin を実行します。
  - リモート・ログイン名を、該当するデータベースのユーザとして追加し、プロシージャを実行するパーミッションを付与します (execute パーミッションが “public” に付与されている場合は、ユーザに特定のパーミッションを付与する必要はありません)。

図 15-1 は、リモート・アクセスができるようにサーバを設定する方法を示します。

図 *15-1:* リモート・プロシージャ・コールを可能にするサーバ設定

ROSE 上のユーザ “joe” が ZINNIA 上のストアド・プロシージャにアクセスする必要がある場合

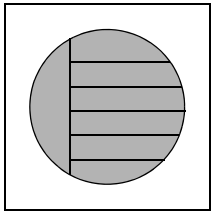
ROSE



sp_addserver ROSE, local
sp_addserver ZINNIA

Interfaces ファイルに ZINNIA のエントリが必要

ZINNIA



sp_addserver ROSE
sp_addlogin joe
sp_addremotelogin ROSE, joe

sp_adduser joe (該当するデータベース上で)
grant execute on *procedure_name* to joe

リモート・サーバの処理に関するオペレーティング・システム固有の情報については、プラットフォームの『インストール・ガイド』を参照してください。

リモート・サーバの管理

表 15-1 は、リモート・サーバの管理に関連する作業と、その作業を実行するために使用するシステム・プロシージャを示します。

表 *15-1:* リモート・サーバの管理に関連する作業

目的	使用	参照箇所
リモート・サーバの追加	sp_addserver	「リモート・サーバの追加」(459 ページ)
リモート・サーバ名の管理	sp_addserver	「リモート・サーバ名の管理」(461 ページ)
サーバ接続オプションの変更	sp_serveroption	「サーバ接続オプションの設定」(461 ページ)
サーバに関する情報の表示	sp_helpserver	「サーバ情報の取得」(463 ページ)
サーバの削除	sp_dropserver	「リモート・サーバの削除」(463 ページ)

リモート・サーバの追加

システム・セキュリティ担当者は、sp_addserver を使用して、sysservers テーブルにエントリを追加します。RPC を発信するサーバでは、ローカル・サーバのエントリ 1 つと、呼び出すリモート・サーバごとに 1 つのエントリを追加する必要があります。

リモート・サーバのエントリを作成するときは、次のどちらかを選択できます。

- *interfaces* ファイルに登録されている名前でリモート・サーバを参照する。
- リモート・サーバに対してローカル名を与える。たとえば、*interfaces* ファイルでの名前が “MAIN_PRODUCTION” であるリモート・サーバに、“main” という名前を付けます。

構文は次のとおりです。

```
sp_addserver lname [{, local | null}
    [, pname]]
```

各パラメータの意味は、次のとおりです。

- *lname* には、リモート・サーバのローカルでの「呼び名」を指定します。この名前が *interfaces* ファイルでのリモート・サーバの名前と同じでない場合は、その名前を 3 番目のパラメータ *pname* として指定する必要があります。

  リモート・サーバは、ローカル・マシン上の *interfaces* ファイルに登録されていなければなりません。登録されていない場合は、リモート・サーバから *interfaces* ファイルのエントリをコピーして、既存の *interfaces* ファイルに追加します。同じポート番号を使用してください。

- local は、このサーバがローカル・サーバとして追加されることを示します。local という値が使用されるのは、起動または再起動の後のみで、ここで指定されたローカル・サーバ名が Adaptive Server の出力メッセージに表示されます。null は、このサーバがリモート・サーバであることを示します。

  注意 ユーザがローカル・サーバから RPC を正常に実行できるようにするには、local オプションを指定してローカル・サーバを追加し、再起動する必要があります。再起動は、グローバル変数 *@@servername* を設定するために必要です。

- *pname* には、リモート・サーバを指定します。これは、*lname* という名前で *interfaces* ファイルに登録されているサーバです。このオプション引数を指定すると、通信する必要がある他の Adaptive Server、Open Server、Backup Server 用のローカル・エイリアスを設定できます。*pname* を指定しない場合のデフォルト値は *lname* になります。

リモート・サーバの追加の例

次の例は DOCS という名前のローカル・サーバ用のエントリを作成します。

```
sp_addserver DOCS, local
```

次の例は GATEWAY という名前のリモート・サーバ用のエントリを作成します。

```
sp_addserver GATEWAY
```

GATEWAY サーバ上で sp_who などのリモート・プロシージャを実行するには、次のいずれかを実行します。

```
GATEWAY.sybsytemprocs.dbo.sp_who
```

または

```
GATEWAY...sp_who
```

次の例は、MAIN_PRODUCTION という名前のリモート・サーバにローカル・エイリアス “main” を与えます。

```
sp_addserver main, null, MAIN_PRODUCTION
```

この場合は、ユーザは次のように入力できます。

```
main...sp_who
```

リモート・サーバ名の管理

master.dbo.sysservers テーブルには、次の 2 つのサーバ名のカラムがあります。

- srvname は、ユーザがリモート・プロシージャ・コールを実行するときに指定するユニークなサーバ名です。
- srvnetname はサーバのネットワーク名であり、*interfaces* ファイル内の名前と一致する必要があります。

サーバをネットワークに追加する場合やネットワークから削除する場合は、sp_addserver を使用して、srvnetname のサーバのネットワーク名を更新します。

たとえば、サーバ MAIN をネットワークから削除して、リモート・アプリケーションを TEMP という名前のサーバに移動するには、次の文を使用すると、ローカル・エイリアスを保持したままネットワーク名を変更できます。

```
sp_addserver MAIN, null, TEMP
```

sp_addserver は、既存のサーバ・エントリのネットワーク名を変更しようとしていることを通知するメッセージを表示します。

サーバ接続オプションの設定

sp_serveroption は、リモート・サーバとの接続に影響を与えるサーバ・オプション timeouts、net password encryption、rpc security model A、rpc security model B を設定します。また、リモート・プロシージャのセキュリティ・モデルを rpc security model B に設定した場合は、sp_serveroption を使用して、オプション security mechanism、mutual authentication、use message confidentiality、use message integrity を設定できます。

sp_serveroption で指定するオプションは、Adaptive Server と Backup Server の間の通信には影響しません。

次の各項では、timeouts、net password encryption、rpc security model A、rpc security model B について説明します。rpc security model B がオンのときに指定できるその他のオプションの詳細については、「リモート・プロシージャのセキュリティ設定」(486 ページ) を参照してください。

timeouts オプションの使用

システム管理者は timeouts オプションを使用して、ローカル・サーバが使用する通常のタイムアウト・コードを無効または有効にできます。

デフォルトでは、timeouts は true に設定され、リモート・ユーザのアクティビティが行われない状態が 1 分間続くと、リモート・ログインを管理するサイト・ハンドラ・プロセスはタイムアウトになります。リモート・プロシージャ・コールに関係するサーバの両方で timeouts が false に設定されている場合は、自動タイムアウトは行われません。timeouts を false に変更するには次のようにします。

```
sp_serveroption GATEWAY, "timeouts", false
```

両方のサーバで timeouts が false に設定された後で、いずれかのサーバから RPC が実行されると、各マシンのサイト・ハンドラはどちらかのサーバが停止するまで動作を続けます。サーバが再び起動されると、オプションは false のままで、ユーザが次に RPC を実行するときにサイト・ハンドラが再設定されます。ユーザが頻繁に RPC を実行する場合は、このオプションを false に設定した方が、システム・リソースの点からは効率的です。物理的接続の設定にはかなりのシステム・オーバヘッドがかかるからです。

net password encryption オプションの使用

システム・セキュリティ担当者は net password encryption オプションを使用して、リモート・サーバとの接続をクライアント側パスワード暗号化ハンドシェイクによって開始するか、または通常の非暗号化パスワード・ハンドシェイク・シーケンスによって開始するかを指定できます。デフォルトは false です。

net password encryption オプションを true に設定すると、処理の順序は次のようになります。

1 ログイン開始パケットが、パスワードなしで送信されます。

2 クライアントは、暗号化が要求されていることをリモート・サーバに知らせます。

3 リモート・サーバは暗号化キーを返します。これは、クライアントがプレーン・テキストのパスワードを暗号化するために使用するキーです。

4 次にクライアントは自分のパスワードを暗号化し、リモート・サーバは渡されたパスワードをこのキーを使って認証します。

次の例は、net password encription を true に設定します。

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

rpc security model オプションの使用

rpc security model A オプションと rpc security model B オプションによって、RPC で使用できるセキュリティの種類を指定します。デフォルトのモデル A を使用する場合は、2 つのサーバ間での暗号化によるメッセージの機密保持などのセキュリティ・サービスはサポートされません。

セキュリティ・モデル B では、ローカルの Adaptive Server はセキュリティ・メカニズムからクレデンシャルを取得し、このクレデンシャルを使用してリモート Adaptive Server との間に安全な物理的接続を確立します。このモデルでは、相互認証、暗号化によるメッセージの機密保持、メッセージ整合性のいずれか 1 つまたは複数のセキュリティ・サービスを使用できます。

サーバ GATEWAY にセキュリティ・モデル A を設定するには、次を実行します。

```
sp_serveroption GATEWAY, "rpc security model A",
   true
```

サーバにセキュリティ・モデル B を設定する方法の詳細については、「リモート・プロシージャのセキュリティ設定」(486 ページ) を参照してください。

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 を使用する場合は、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。サーバ名を指定して sp_helpserver を使用すると、そのサーバの情報だけが表示されます。

sp_helpserver [*server*]

sp_helpserver は、master..sysremotelogins テーブル内の srvname と srvnetname の両方をチェックします。

リモート・サーバの設定に関するオペレーティング・システム固有の情報については、プラットフォームの『インストール・ガイド』を参照してください。

リモート・サーバの削除

システム・セキュリティ担当者は、sp_dropserver システム・プロシージャを使用して、sysservers からサーバを削除できます。

sp_dropserver *server* [, droplogins]

各パラメータの意味は、次のとおりです。

- *server* は、削除するサーバの名前です。
- droplogins を使用すると、リモート・サーバと、そのサーバのリモート・ログイン情報すべてを削除できます。droplogins オプションを使用しない場合は、リモート・ログインが関連付けられているサーバを削除することはできません。

次の文は、GATEWAY サーバと、このサーバに関連付けられているすべてのリモート・ログインを削除します。

```
sp_dropserver GATEWAY, droplogins
```

ローカル・サーバの削除には droplogins を使用する必要はありません。ローカル・サーバに関連付けられているリモート・ログイン情報はないためです。

リモート・ログインの追加

Adaptive Server のシステム・セキュリティ担当者とシステム管理者は、そのサーバにアクセスできるリモート・ユーザとそのリモート・ユーザの ID の管理を分担します。システム管理者は sp_addremotelogin を使ってリモート・ログインを追加し、sp_dropremotelogin を使ってリモート・ログインを削除します。システム・セキュリティ担当者は、sp_remoteoption を使って、パスワードの検査が必要かどうかを管理します。

ユーザのサーバ ID のマッピング方法

リモート・サーバからのログインをローカル・サーバにマップするには、次の 3 つの方法があります。

- 特定のリモート・ログインを特定のローカル・ログイン名にマップします。たとえば、リモート・サーバ上のユーザ “joe” を、“joesmith” にマップします。
- 1 つのリモート・サーバからのすべてのログインを、1 つのローカル名にマップします。たとえば、MAIN サーバからリモート・プロシージャ・コールを送信するすべてのユーザを “remusers” にマップします。
- 1 つのリモート・サーバからのすべてのログインが各自のリモート・ログイン名を使用するようにします。

最初のオプションは他の 2 つのオプションと組み合わせることができ、その個別のマッピングは他の 2 つの全体的なマッピングに優先します。2 番目と 3 番目のオプションは互いに排他的です。どちらか一方だけを使用することはできますが、両方を使用することはできません。

マッピング・オプションの変更

sp_dropremotelogin を使用して、古いマッピングを削除します。

sp_addremotelogin を使用して、リモート・ログインを追加します。

```
sp_addremotelogin remoteserver [, loginame
    [, remotename]]
```

ローカル名が master..syslogins に登録されていない場合は、sp_addlogin を使って Adaptive Server ログインとして追加してから、リモート・ログインを追加してください。

sp_addremotelogin を実行できるのは、システム管理者だけです。『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。

リモート・ログインを特定のローカル名にマップする方法

次の例は、リモート・システムの “pogo” という名前のログインを、ローカル・ログイン名 “bob” にマップします。このユーザは “pogo” としてリモート・システムにログインします。ユーザ “pogo” が GATEWAY からリモート・プロシージャ・コールを実行するとき、ローカル・システムはリモート・ログイン名を “bob” にマップします。

```
sp_addlogin bob
sp_addremotelogin GATEWAY, bob, pogo
```

すべてのリモート・ログインを 1 つのローカル名にマップする方法

次の例は、すべてのリモート・ログイン名をローカル名 “albert” にマップするエントリを作成します。前の項で説明したように、個別のマッピングを持つ名前以外のすべての名前が “albert” にマップされます。たとえば、“pogo” を “bob” にマップして、それ以外のすべてのログインを “albert” にマップした場合も、“pogo” は “bob” にマップされたままになります。

```
sp_addlogin albert
sp_addremotelogin GATEWAY, albert
```

sp_addremotelogin を使用してリモート・サーバのすべてのユーザを同じローカル名にマップする場合は、sp_remoteoption を使用して、それらのユーザに “trusted” オプションを指定します。たとえば、“albert” にマップされるサーバ GATEWAY のユーザすべてを trusted にするには、次のように指定します。

```
sp_remoteoption GATEWAY, albert, NULL, trusted, true
```

trusted が指定されていないログインは、そのローカル・サーバで RPC を実行することはできません。ただし、リモート・サーバへのログイン時に、ローカル・サーバのパスワードを指定すれば、ローカル・サーバ上で RPC を実行できます。これらのユーザは、Open Client Client-Library を使用するときに、ct_remote_pwd ルーチンを使用してサーバ対サーバ接続のパスワードを指定することができます。isql と bcp は、ユーザが RPC 接続用のパスワードを指定することを許可しません。sp_remoteoption の詳細については、「リモート・ユーザのパスワードの検査」(468 ページ) を参照してください。

警告！ 複数のリモート・ログインを 1 つのローカル・ログインにマップしないでください。サーバ上での個人の責任が不明確になります。監査対象のアクションの追跡によって特定できるのはローカル・サーバのログインだけであり、リモート・サーバ上の個々のログインを特定することはできません。

ネットワークベース・セキュリティの使用

ユーザが「統一化ログイン」を使用してリモート・サーバにログインする場合は、そのログインにはローカル・サーバでも trusted が指定されている必要があります。そうでない場合は、ユーザはリモート・サーバにログインするときにサーバのパスワードを指定する必要があります。

警告！ sp_remoteoption の trusted モードを使用すると、このような trusted ユーザのパスワードが検証されないので、サーバのセキュリティは低くなります。

ローカル・サーバのリモート・ログイン名の保持

リモート・ユーザが、ローカル・サーバの使用時にリモート・ログイン名を保持できるようにするには、次の手順に従います。

1 sp_addlogin を使用して、リモート・サーバからの各ログインに対するログインを作成します。

2 sp_addremotelogin を使用して、サーバ全体に対する 1 つのエントリを master..sysremotelogins に作成します。リモート・ログイン名の値は null、suid の値は -1 とします。次に例を示します。

```
sp_addremotelogin GATEWAY
```

リモート・ユーザ・ログインのマッピング例

次の文は、master..sysservers に記録されているローカル・サーバとリモート・サーバの情報を表示します。

```
select srvid, srvname from sysservers
srvid  srvname
-----  -----------
    0  SALES
    1  CORPORATE
    2  MARKETING
    3  PUBLICATIONS
    4  ENGINEERING
```

SALES サーバはローカル・サーバです。それ以外のサーバは、リモート・サーバです。

次の文は、master..sysremotelogins に保管されているリモート・サーバとユーザに関する情報を表示します。

```
select remoteserverid, remoteusername, suid
  from sysremotelogins
remoteserverid     remoteusername    suid
---------------    --------------    ------
1                  joe               1
1                  nancy             2
1                  NULL              3
3                  NULL              4
4                  NULL              -1
```

この結果における remoteserverid の値を、前の結果の srvid の値と照合すれば、remoteusername がどのサーバで有効であるかがわかります。たとえば最初の結果では、srvid 1 は、CORPORATE サーバを示します。2 番目の結果では、remoteserverid 1 は、同じサーバを示します。したがって、リモート・ユーザ・ログイン名 “joe” と “nancy” は、CORPORATE サーバ上で有効です。

次の文は、master..syslogins のエントリを表示します。

```
select suid, name from syslogins
suid     name
------   ------------
    1    sa
    2    vp
    3    admin
    4    writer
```

この 3 つのクエリの結果から、次のことがわかります。

- リモートの CORPORATE サーバ (srvid と remoteserverid が 1) 上のリモート・ユーザ名 “joe” (suid 1) は、“sa” ログイン (suid 1) にマップされます。
- リモートの CORPORATE サーバ (srvid と remoteserverid が 1) 上のリモート・ユーザ名 “nancy” (suid 2) は、“vp” ログイン (suid 2) にマップされます。
- CORPORATE サーバの他のログイン (remoteusername “NULL” は、“admin” ログイン (suid 3) にマップされます。
- PUBLICATIONS サーバ (srvid と remoteserverid が 3) のログインはすべて “writer” ログイン (suid 4) にマップされます。
- ENGINEERING サーバ (srvid と remoteserverid が 4) のすべてのログインは、そのリモート・ユーザ名 (suid -1) によって master..syslogins での検索が行われます。
- sysremotelogins には、remoteserverid が MARKETING サーバを示すエントリはありません。したがって、MARKETING サーバにログインしたユーザが、そのサーバからリモート・プロシージャ・コールを実行することはできません。

リモート・ユーザのマップの手順と個々のストアド・プロシージャに対してパーミッションを設定する機能によって、どのリモート・ユーザがローカル・プロシージャにアクセスできるかを管理できます。たとえば、CORPORATE サーバからの “vp” ログインが特定のローカル・プロシージャを実行できるようにして、それ以外の CORPORATE のすべてのログインについては “admin” ログインがパーミッションを所有しているプロシージャを実行できるようにします。

注意 通常は、リモート・サーバのユーザのパスワードはローカル・サーバのパスワードと一致していなければなりません。

リモート・ユーザのパスワードの検査

システム・セキュリティ担当者は、sp_remoteoption を使用して、リモート・ユーザがローカル・サーバにログインするときにパスワードの検査を行うかどうかを設定できます。デフォルトでは、パスワードを検査します (“untrusted” モード)。trusted モードでは、ローカル・サーバはログイン・アカウントについてのユーザ・アクセスの確認を行わずに、他のサーバやフロントエンド・アプリケーションからのリモート・ログインを許可します。

引数を指定して sp_remoteoption を実行すると、指定したユーザのモードを変更できます。

```
sp_remoteoption [remoteserver, loginame, remotename,
    optname, {true | false}]
```

次の例は、ユーザ “bob” に trusted モードを設定します。

```
sp_remoteoption GATEWAY, pogo, bob, trusted,
    true
```

untrusted モードを使用した場合の影響

untrusted モードを使用した場合の影響は、ユーザのクライアント・プログラムによって異なります。isql や一部のユーザ・アプリケーションでは、リモート・サーバとローカル・サーバでログインのパスワードが同じでなければなりません。Open Client アプリケーションの場合は、ローカル・ログインのパスワードがサーバごとに異なっていてもかまわないこともあります。

“untrusted” モードでパスワードを変更するには、アクセスするすべてのリモート・システム上でパスワードを変更してから、ローカル・サーバのパスワードを変更してください。ローカル・サーバのパスワードを先に変更すると、リモート・プロシージャ・コールを発行してリモート・サーバ上で sp_password を実行するときに、パスワードが一致しないことになります。

リモート・サーバ上のパスワードを変更する構文は次のとおりです。

remote_server...sp_password *caller_passwd*, *new_passwd*

ローカル・サーバでの構文は次のとおりです。

sp_password *caller_passwd*, *new_passwd*

詳細については、「パスワードの変更」(406 ページ) を参照してください。

リモート・ログイン情報の取得

sp_helpremotelogin は、サーバ上のリモート・ログインについての情報を表示します。次の例では、リモート・ログイン “pogo” は、ローカルではログイン名 “bob” にマッピングされていることがわかります。他のリモート・ログインはすべてリモート名をそのまま使います。

```
sp_helpremotelogin

server      remote_user_name      local_user_name       options
---------   -----------------     -----------------     ---------
GATEWAY     **mapped locally**    **use local name**    untrusted
GATEWAY     pogo                  bob                   untrusted
```

リモート・ログインの設定パラメータ

表 15-2 は、RPC に影響を与える設定パラメータを示します。これらの設定パラメータはすべて sp_configure を使用して設定されます。また、Adaptive Server を再起動しないと有効になりません。

表 *15-2: RPC* に影響を与える設定パラメータ

設定パラメータ	デフォルト
allow remote access	1
number of remote logins	20
number of remote sites	10
number of remote connections	20
remote server pre-read packets	3

個々の設定パラメータの詳細については、「第 5 章 設定パラメータ」を参照してください。

第 16 章

外部認証

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

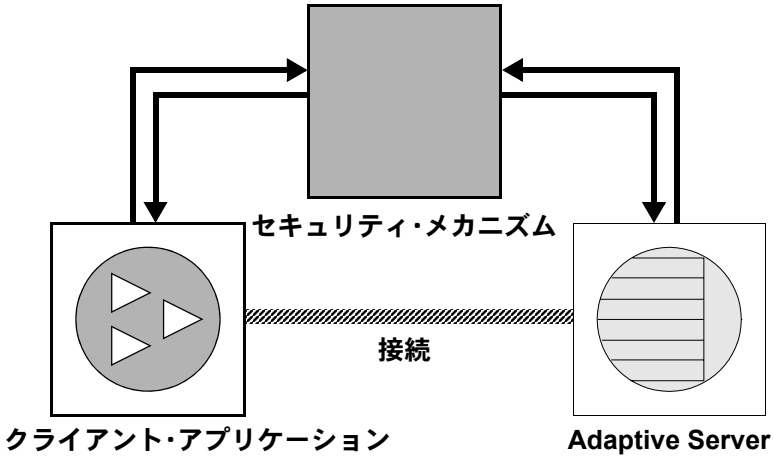
大規模な異機種アプリケーションでは、ログインを集中レポジトリで認証することによってセキュリティを強化できます。Adaptive Server では、次の外部認証メソッドがサポートされています。

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

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

図 16-1 は、セキュリティ・メカニズムを使用して Adaptive Server とのセキュア接続を確立するクライアント・アプリケーションを示しています。

図 *16-1: クライアントと Adaptive Server* の間のセキュア接続の確立



クライアント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場合は、次の処理が行われます。

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

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

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

クライアントがメッセージ保護サービスを要求する場合は、次の処理が行われます。

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

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

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

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

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

セキュリティ・サービスと Adaptive Server

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

- 統一化ログイン － ユーザを一度だけ認証する。ユーザは、Adaptive Server にログインするたびに名前とパスワードを入力する必要はない。
- メッセージの機密保持 － ネットワーク上で転送されるデータを暗号化する。
- 相互認証 － クライアントとサーバの身元を検証する。相互認証を要求できるのはクライアントのみです。Adaptive Server は相互認証を要求できません。
- メッセージ整合性 － データ通信が変更されていないことを検証する。
- リプレイの検出 － データが侵入者によって傍受されていないことを確認する。
- 順序不整合の検査 － データ通信の順序を確認する。
- メッセージ・オリジンの検査 － メッセージのオリジンを確認する。
- リモート・プロシージャ・セキュリティ － 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かぎりません。「使用できるセキュリティ・サービスの情報の取得」(494 ページ) を参照してください。

ネットワークベース・セキュリティの管理

表 16-1 は、Adaptive Server のネットワークベース・セキュリティ機能を使用するための全体的なプロセスを示します。Adaptive Server をインストールしてから、表 16-1 の手順を実行します。

表 *16-1:* ネットワークベース・セキュリティの管理

手順	説明	参照箇所
1. 次の設定ファイルを設定する。 • *libtcl.cfg* • *objectid.dat* • *interfaces* (またはディレクトリ・サービス)	*libtcl.cfg* ファイルを編集する。*objectid.dat* ファイルを編集する。*interfaces* ファイルまたはディレクトリ・サービスを編集する。	• 「セキュリティの設定ファイルの設定」(475 ページ) • 使用しているプラットフォームの『Open Client/Server 設定ガイド』
2. セキュリティ・メカニズムのセキュリティ管理者によって各ユーザおよび Adaptive Server と Backup Server 用のログインが作成されていることを確認する。	セキュリティ管理者は、ユーザとサーバの名前とパスワードをセキュリティ・メカニズムに追加する必要がある。 分散コンピューティング環境 (DCE) を使用する場合は、セキュリティ管理者はサーバ・エントリに対する *keytab* ファイルを作成する必要がある。	• 使用しているセキュリティ・メカニズムのマニュアル • 「セキュリティ・メカニズムに対するユーザとサーバの識別」(480 ページ)
3. インストール環境にセキュリティを設定する。	sp_configure を使用する。	「Adaptive Server でのセキュリティの設定」(481 ページ)
4. Adaptive Server を再起動する。	use security services パラメータをアクティブにする。	使用しているプラットフォームの『設定ガイド』
5. 企業全体のログインをサポートするためのログインを Adaptive Server に追加する。	sp_addlogin を使用してユーザを追加する。必要であれば、sp_configure を使用してデフォルト・セキュア・ログインを指定する。	「統一化ログインをサポートするためのログインの追加」(484 ページ)
6. リモート・プロシージャ用のセキュリティ・モデルを決定し、ローカル・サーバとリモート・サーバに RPC セキュリティを設定する。	sp_serveroption を使用し、セキュリティ・モデル A または B を選択する。	「リモート・プロシージャのセキュリティ設定」(486 ページ)
7. サーバに接続し、セキュリティ・サービスを使用する。	isql_r や Open Client Client-Library を使い、使用するセキュリティ・サービスを指定して、Adaptive Server に接続する。	• 「サーバへの接続とセキュリティ・サービスの使用」(491 ページ) • 使用しているプラットフォームの『Open Client/Server 設定ガイド』 • 『Open Client Client-Library/C リファレンス・マニュアル』の「セキュリティ機能」
8. 利用できるセキュリティ・サービスとセキュリティ・メカニズムをチェックする。	show_sec_services 関数および is_sec_services_on 関数を使用し、利用できるセキュリティ・サービスをチェックする。 Adaptive Server がサポートしているセキュリティ・メカニズムとそのセキュリティ・サービスのリストについては、select を使用して syssecmechs システム・テーブルを問い合わせる。	「使用できるセキュリティ・サービスの情報の取得」(494 ページ)

セキュリティの設定ファイルの設定

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

表 *16-2:* 設定ファイルの名前とロケーション

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

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

サーバのセキュリティ情報の指定

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

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

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

セキュリティ・メカニズムを指定する UNIX ツール

使用するセキュリティ・メカニズムは、次のように指定します。

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

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

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

サーバの属性を指定するデスクトップ・ツール

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

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

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

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

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

ネットワーク・ドライバのエントリ

ネットワーク・ドライバ・エントリの構文は、次のとおりです。

driver=protocol description

パラメータの意味は次のとおりです。

- *driver* － ネットワーク・ドライバの名前。
- *protocol* － ネットワーク・プロトコルの名前。
- *description* － エントリの説明。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

パラメータの意味は次のとおりです。

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

  デフォルトのローカル名は次のとおりです。

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

  デフォルト以外のローカル・メカニズム名を使用する場合は、objectid.dat ファイルにあるローカル名を変更します (「objectid.dat ファイル」(479 ページ) の例を参照)。
- *driver* － セキュリティ・ドライバの名前。UNIX プラットフォームのすべてのドライバのデフォルト・ロケーションは、*$SYBASE/$SYBASE_OCS/config*。Windows プラットフォームのデフォルト・ロケーションは、*%SYBASE%¥%SYBASE_OCS%¥dll*。

- *init-string*　はドライバの初期化文字列です。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る。
  - DCE ドライバの場合の *init-string* の構文は次のとおり。ただし、*cell_name* は DCE セルの名前。

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

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

UNIX プラットフォーム情報

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

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

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

これらのセクションを特定の順序に並べる必要はありません。

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

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

Sun Solaris の *libtcl.cfg* の例

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

[DIRECTORY]
;dce=libsybddce.so ditbase=/.:/subsys/sybase/dataservers
;dce=libsybddce.so ditbase=/.:/users/cfrank

[SECURITY]
dce=libsybsdce.so secbase=/.../svrsole4_cell
```

この *libtcl.cfg* ファイルは、DCE セキュリティ・サービスを使用します。[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、DCE サービスを表す 1.3.6.1.4.1.897.4.6.1 などのグローバル・オブジェクト識別子を “dce” などのローカル名にマッピングします。objectid.dat ファイルには、文字セット用の [CHARSET] セクションや、セキュリティ・サービス用の [SECURITY] セクションが含まれています。objectid.dat ファイルの例を次に示します。

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

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

たとえば、libtcl.cfg に次のセクションがあるとします。

```
[SECURITY]
dce=libsybsdce.so secbase=/.../svrsole4_cell
```

上記を次のように変更します。

```
[SECURITY]
dce_group=libsybsdce.so secbase=/.../svrsole4_cell
```

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

```
1.3.6.1.4.1.897.4.6.1    = dce_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 (ユーザとサーバの両方) を定義する必要があります。表 16-3 は、ユーザとサーバの追加に使用できるツールのリストです。

表 *16-3:* セキュリティ・メカニズムに対するユーザとサーバの定義

セキュリティ・メカニズム	コマンドまたはツール
DCE	新しいプリンシパル (ユーザまたはサーバ) を作成するには、dcecp の user コマンドと create コマンドを使用する。さらに、keytab create コマンドを使用して、プリンシパルのパスワードが暗号化された形式で格納された DCE keytab ファイルを作成する。 DCE に対してサーバを定義するときに、新しいプリンシパルがサーバとして動作するようにコマンド・オプションで指定する。
Kerberos	ユーザとサーバの定義方法については、Kerberos のベンダ固有のツールを参照。Kerberos と Adaptive Server の詳細については、「Kerberos の使用」(495 ページ) を参照。
Windows NT LAN Manager	[ユーザー マネージャ] ツールを実行し、Windows NT LAN Manager にユーザを定義する。Adaptive Server の名前を Windows NT LAN Manager にユーザとして定義し、Adaptive Server をそのユーザ名で表示する。

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

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

Adaptive Server でのセキュリティの設定

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

ネットワークベース・セキュリティの有効化

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

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

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

Adaptive Server が特定のクライアントに使用するセキュリティ・メカニズムを決定する方法については、「クライアントへのセキュリティ・メカニズムの使用」(493 ページ) を参照してください。

統一化ログインの要求

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

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

たとえば、すべてのログインがセキュリティ・メカニズムによって認証されるように要求するには、次のコマンドを実行します。

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

セキュア・デフォルト・ログインの確立

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

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

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

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

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

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

```
sp_addlogin gen_auth, pwgenau
```

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

2 次のように入力して、ログインをセキュア・デフォルトとして指定します。

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

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

注意 このセキュア・デフォルト・ログインに関連付けられている suid は複数のユーザによって使用されます。したがって、デフォルト・ログインによるすべてのアクティビティに対して監査を行うように設定することをおすすめします。また、sp_addlogin 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。

詳細については、「統一化ログインをサポートするためのログインの追加」(484 ページ) と「Adaptive Server へのログインの追加」(381 ページ) を参照してください。

セキュリティ・メカニズムのログイン名からサーバ名へのマッピング

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

表 16-4 は、ログイン名に使用されている無効な文字を Adaptive Server が変換する方法を示します。

表 *16-4:* ログイン名の無効な文字の変換

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

暗号化によるメッセージの機密保持の要求

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

sp_configure *configuration_parameter*, [0 | 1]

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

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

データ整合性の要求

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

ネットワークベース・セキュリティのメモリ要件

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

構文は次のとおりです。

```
sp_configure "max total_memory" value
```

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

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

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

統一化ログインをサポートするためのログインの追加

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

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

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

このため、管理者は、有効なログインとして定義されているユーザだけに Adaptive Server の使用を許可するか、ユーザがデフォルト・ログインを使用してログインできるようにするかを決める必要があります。デフォルトを定義するには、デフォルト・ログインを master..syslogins に追加し、sp_configure を使用します。「セキュア・デフォルト・ログインの確立」(481 ページ) を参照してください。

ログインを追加するための一般的な手順

サーバにログインを追加したり、オプションで、ユーザに 1 つ以上のデータベースに対する適切な役割や権限を追加したりするには、表 16-5 に記載されている一般的な手順に従います。

表 *16-5:* ログインの追加とデータベースへのアクセスの許可

作業	必要な役割	コマンドまたはプロシージャ	参照箇所
1. ユーザに対応するログインを追加する。	システム・セキュリティ担当者	sp_addlogin	「Adaptive Server へのログインの追加」(381 ページ)
2. ユーザを 1 つ以上のデータベースに追加する。	システム管理者またはデータベース所有者	sp_adduser － データベース内からこのプロシージャを実行する。	「データベースへのユーザの追加」(384 ページ)
3. ユーザをデータベースのグループへ追加する。	システム管理者またはデータベース所有者	sp_changegroup － データベース内からこのプロシージャを実行する。	• 「ユーザのグループ・メンバシップの変更」(408 ページ) • 『ASE リファレンス・マニュアル』の「sp_changegroup」
4. システム標準の役割をユーザに付与する。	システム管理者またはシステム・セキュリティ担当者	grant role	• 「ユーザに対する役割の作成と割り当て」(390 ページ) • 『ASE リファレンス・マニュアル』の「grant」
5. ユーザ定義の役割を作成し、作成した役割をユーザに付与する。	システム・セキュリティ担当者	create role grant role	• 『ASE リファレンス・マニュアル』の「ユーザに対する役割の作成と割り当て」(390 ページ) • 『ASE リファレンス・マニュアル』の「grant」 • 『ASE リファレンス・マニュアル』の「create role」
6. データベース・オブジェクトへのアクセス権を与える。	データベース・オブジェクト所有者		「第 17 章 ユーザ・パーミッションの管理」

リモート・プロシージャのセキュリティ設定

Adaptive Server は、他のサーバに接続して RPC (リモート・プロシージャ・コール) を実行するときにクライアントとして動作します。

1 つの「物理接続」が 2 つのサーバ間で確立します。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。

セキュリティ・モデル A

デフォルトのセキュリティ・モデル A では、2 つのサーバ間での暗号化によるメッセージの機密保持などのセキュリティ・サービスはサポートされません。

セキュリティ・モデル B

セキュリティ・モデル B では、ローカルの Adaptive Server はセキュリティ・メカニズムからクレデンシャルを受信し、このクレデンシャルを使用してリモート Adaptive Server との間に安全な物理的接続を確立します。モデル B では、以下のセキュリティ・サービスを使用できます。

- 相互認証 − ローカル・サーバはリモート・サーバのクレデンシャルを取り出し、それをセキュリティ・メカニズムで検証して、リモート・サーバを認証します。両方のサーバのクレデンシャルが認証され、検証されます。
- 暗号化によるメッセージの機密保持 − メッセージはリモート・サーバに送信されるときに暗号化され、リモート・サーバからの結果も暗号化されます。
- メッセージの整合性 − サーバ間のメッセージは勝手に変更されないようチェックされます。

統一化ログインとリモート・プロシージャ・モデル

ローカル・サーバとリモート・サーバにセキュリティ・サービスを使用するように設定すると、サーバのセキュリティ・モデルがどちらであっても、以下の 2 つのいずれかの方法で両方のサーバに統一化ログインによるログインが可能です。

- システム・セキュリティ担当者は、リモート・サーバで sp_remoteoption を使用してユーザを “trusted” と定義する。DCE などのセキュリティ・メカニズムがユーザとパスワードを認証する。ユーザは「統一化ログイン」を使用してローカル・サーバにアクセスし、リモート・サーバで RPC を実行する。ユーザはリモート・サーバで信頼されている (trusted) ため、パスワードを入力する必要がない。
- ユーザは、ローカル・サーバに接続するときにリモート・サーバのパスワードを指定する。リモート・サーバのパスワードを指定するしくみとして、Open Client Client-Library/C に ct_remote_pwd ルーチンが用意されている。『Open Client Client-Library/C リファレンス・マニュアル』を参照のこと。

RPC でのセキュリティ・モデルの設定

RPC のセキュリティ・モデルをモデル A 用およびモデル B 用に設定するには、sp_serveroption を使用します。構文は次のとおりです。

sp_serveroption *server*, *optname*, [true | false]

セキュリティ・モデルを設定するには、*optname* を rpc security model A または rpc security model B に設定します。*server* はリモート・サーバ名です。

たとえば、リモート・サーバ TEST3 でセキュリティ・モデル B を使用するように設定するには、次のコマンドを実行します。

```
sp_serveroption test3, "rpc security model B", true
```

モデル A の場合、サーバ・オプションを設定する必要はありません。

『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。

RPC にセキュリティ・モデル B を設定する規則

RPC にセキュリティ・モデル B を使用するように設定するときは、以下の規則に従います。

- サーバは、両方ともセキュリティ・モデル B を使用する必要がある。
- 両方のサーバが同じセキュリティ・メカニズムを使用する必要がある。そのセキュリティ・メカニズムは、sp_serveroption を使用して設定されたセキュリティ・サービスをサポートするものでなければならない。
- ローカル・サーバのシステム・セキュリティ担当者は、リモート・サーバが必要とするセキュリティ・サービスを指定する必要がある。たとえば、すべてのメッセージにメッセージ機密保持セキュリティ・サービスを使用するようにリモート・サーバから要求される場合は、システム・セキュリティ担当者は sp_serveroption を使用して use message confidentiality をアクティブにする必要がある。
- セキュリティ・メカニズムによって認証済みのログインが「統一化ログイン」を使用して Adaptive Server にログインした場合に、そのログインがリモート・サーバ上で RPC を実行するには、リモート・サーバ上でそのログインに “trusted” が指定されているか、そのログインがリモート・サーバのパスワードを指定することが必要となる。Open Client Client-Library を使用するユーザは、ルーチン ct_remote_pwd を使用してサーバ間の接続のパスワードを指定できる。Adaptive Server のシステム管理者は、信頼されている (trusted) ユーザがパスワードを入力しないでリモート・サーバを使用できるように sp_remoteoption を使用して設定できる。

RPC にセキュリティ・モデル B を使用する準備

表 16-6 は、セキュリティ・モデル B を使用して RPC のセキュリティを設定する手順を示します。

表 *16-6: RPC* でのセキュリティ・モデル *B* の使用

実行者、タスク、状況	コマンド、システム・プロシージャ、ツール	参照箇所
システム管理者がオペレーティング・システムから： 1. *interfaces* ファイルまたはディレクトリ・サービスに、両方のサーバのエントリと、セキュリティ・メカニズムを指定した secmech 行が含まれていることを確認する。	UNIX の場合：dscp デスクトップの場合：dsedit	「サーバのセキュリティ情報の指定」(475 ページ) 『Open Client/Server 設定ガイド UNIX 版』の「dscp」 『Open Client/Server 設定ガイド Windows 版』の「dsedit」
システム・セキュリティ担当者がリモート・サーバで： 2. ローカル・サーバを master..sysservers に追加する。	sp_addserver 例： `sp_addserver "lcl_server"`	「リモート・サーバの追加」(459 ページ) 『ASE リファレンス・マニュアル』の「sp_addserver」
システム・セキュリティ担当者がリモート・サーバで： 3. ログインを master..syslogins に追加する。	sp_addlogin 例： `sp_addlogin user1, "pwuser1"`	「Adaptive Server へのログインの追加」(381 ページ) 『ASE リファレンス・マニュアル』の「sp_addlogin」

システム・セキュリティ担当者がリモート・サーバで：
4. use security services をオンに設定し、rpc security model B をローカル・サーバとの接続モデルとして設定する。
	sp_configure (use security services を設定するため)
	sp_serveroption (RPC セキュリティ・モデルを設定するため)
	例：
	`sp_configure "use security services", 1`
	`sp_serveroption lcl_server, "rpc security model B", true`
		「RPC でのセキュリティ・モデルの設定」(487 ページ)
		「ネットワークベース・セキュリティの有効化」(481 ページ)
		「第 5 章 設定パラメータ」の「use security services」
		『リファレンス・マニュアル：プロシージャ』の「sp_configure」と「sp_serveroption」

システム管理者がリモート・サーバで： 5. 必要に応じて、特定のユーザを “trusted” と指定し、パスワードを入力しないでローカル・サーバからリモート・サーバにログインできるようにする。	sp_remoteoption 例： `sp_remoteoption lcl_server, user1, user1, trusted, true`	「リモート・ユーザのパスワードの検査」(468 ページ) 『リファレンス・マニュアル：プロシージャ』の「sp_remoteoption」

システム・セキュリティ担当者がローカル・サーバで：
6. ローカル・サーバとリモート・サーバの両方を master..sysservers に追加する。
	sp_addserver
	例：
	`sp_addserver lcl_server, local`
	`sp_addserver rem_server`
		「リモート・サーバの追加」(459 ページ)
		『ASE リファレンス・マニュアル』の「sp_addserver」

実行者、タスク、状況	コマンド、システム・プロシージャ、ツール	参照箇所
システム・セキュリティ担当者がローカル・サーバで： 7. ログインを master..logins に追加する。	sp_addlogin 例：`sp_addlogin user1, "pwuser1"`	「Adaptive Server へのログインの追加」(381 ページ) 『ASE リファレンス・マニュアル』の「sp_addlogin」

システム・セキュリティ担当者がローカル・サーバで：
8. use security services をオンに設定し、rpc security model B をリモート・サーバとの接続モデルとして設定する。
	sp_configure (use security services を設定するため)
	sp_serveroption (RPC セキュリティ・モデルを設定するため)
	例：
	`sp_configure "use security services", 1`
	`sp_serveroption rem_server, "rpc security model B", true`
		「RPC でのセキュリティ・モデルの設定」(487 ページ)
		「ネットワークベース・セキュリティの有効化」(481 ページ)
		「第 5 章 設定パラメータ」の「use security services」
		『リファレンス・マニュアル：プロシージャ』の「sp_configure」と「sp_serveroption」

システム・セキュリティ担当者がローカル・サーバで：
9. リモート・サーバとの接続に使用するセキュリティ・メカニズムとセキュリティ・サービスを指定する。
	sp_serveroption
	例：
	`sp_serveroption rem_server, "security mechanism", dce`
	`sp_serveroption rem_server, "use message integrity", true`
		「サーバ接続オプションの設定」(461 ページ)
		『リファレンス・マニュアル：プロシージャ』の「sp_serveroption」

RPC にセキュリティ・モデル B を使用する場合の設定例

この例では、以下を前提とします。

- ローカル・サーバ “lcl_serv” が、リモート・サーバ “rem_serv” で RPC を実行する。
- サーバは両方とも、セキュリティ・モデル B と DCE セキュリティ・サービスを使用する。
- RPC セキュリティ・サービスのうち、相互認証とメッセージ整合性を使用する。
- “user1” と “user2” が統一化ログインを使用してローカル・サーバ “lcl_serv” にログインし、“rem_serv” で RPC を実行する。これらのユーザは “rem_serv” 上では “trusted” と定義されるので、リモート・サーバのパスワードを入力する必要はない。
- “user3” は統一化ログインを使用せず、trusted と定義されないため、Adaptive Server にログインするときにパスワードを入力する必要がある。

サーバ間での RPC のセキュリティを設定するには、次の手順に従います。

interfaces ファイルまたはディレクトリ・サービスに、“rem_serv” および “lcl_serv” のエントリが必要です。各エントリに、“dce” セキュリティ・サービスを指定する必要があります。たとえば、dscp ユーティリティを使用した場合は、*interfaces* ファイルに次のようなエントリが作成されます。

```
## lcl_serv (3201)
lcl_serv
master tli tcp /dev/tcp ¥x00020c8182d655110000000000000000
query tli tcp /dev/tcp ¥x00020c8182d655110000000000000000
secmech 1.3.6.1.4.1.897.4.6.1
## rem_serv (3519)
rem_serv
master tli tcp /dev/tcp ¥x000214ad82d655110000000000000000
query tli tcp /dev/tcp ¥x000214ad82d655110000000000000000
secmech 1.3.6.1.4.1.897.4.6.1
```

リモート・サーバ “rem_serv” のシステム・セキュリティ担当者が次のコマンドを実行します。

```
sp_addserver 'lcl_serv'
sp_addlogin user1, "eracg12"
sp_addlogin user2, "esirpret"
sp_addlogin user3, "drabmok"
sp_configure "use security services", 1
sp_serveroption lcl_serv, "rpc security model B", true
sp_serveroption lcl_serv, "security mechanism", dce
```

リモート・サーバ “rem_serv” のシステム管理者が次のコマンドを実行します。

```
sp_remoteoption lcl_serv, user1, user1, trusted, true
sp_remoteoption lcl_serv, user2, user2, trusted, true
```

ローカル・サーバ “lcl_serv” のシステム・セキュリティ担当者が次のコマンドを実行します。

```
sp_addserver lcl_serv, local
sp_addserver rem_serv
sp_addlogin user1, "eracg12"
sp_addlogin user2, "esirpret"
sp_addlogin user3, "drabmo1"
sp_configure "use security services", 1
sp_configure rem_serv, "rpc security model B", true
sp_serveroption rem_serv, "security mechanism", dce
sp_serveroption rem_serv, "mutual authentication" true
sp_serveroption rem_serv, "use message integrity" true
```

注意 これらのサーバでセキュリティ・サービスを使用するには、サーバを再起動して静的パラメータ use security services を有効にする必要があります。

リモート・サーバ情報の取得

sp_helpserver は、サーバに関する情報を表示します。引数を指定しないで sp_helpserver を実行すると、sysservers に登録されているすべてのサーバについての情報が表示されます。特定のサーバを指定すると、そのサーバに関する情報を表示できます。構文は次のとおりです。

```
sp_helpserver [server]
```

たとえば、GATEWAY サーバに関する情報を表示するには、次のコマンドを実行します。

```
sp_helpserver GATEWAY
```

サーバへの接続とセキュリティ・サービスの使用

isql ユーティリティと bcp ユーティリティでは、以下のコマンドライン・オプションを使用することにより、その接続でネットワークベース・セキュリティ・サービスを有効にすることができます。

- -K *keytab_file*
- -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。

- -K *keytab_file* は、DCE セキュリティの場合にのみ使用できます。このオプションでは、ユーザがサーバにログインするためのセキュリティ・キーが格納されている DCE keytab ファイルを指定します。keytab ファイルは、DCE の dcecp ユーティリティを使用して作成します。DCE のマニュアルを参照してください。

  -K オプションを指定しないで isql を実行する場合は、ユーザは DCE にログインする必要があります。-U オプションを指定する場合、-U で指定する名前は、DCE でそのユーザに定義された名前と同じでなければなりません。

- -R *remote_server_principal* は、セキュリティ・メカニズムに対して定義されているサーバのプリンシパル名を指定します。デフォルトでは、サーバのプリンシパル名はサーバのネットワーク名 (-S オプションまたは DSQUERY 環境変数で指定) と一致します。サーバのプリンシパル名とネットワーク名が同じでない場合は、-R オプションを使用する必要があります。

- -V *security_options* は、ネットワークベースのユーザ認証を指定します。このオプションを使用する場合、ユーザはユーティリティを実行する前にネットワークのセキュリティ・システムにログインする必要があります。この場合に、-U オプションを指定するのであれば、セキュリティ・メカニズムに対して定義されているネットワーク・ユーザ名を入力する必要があります。-P オプションで指定したパスワードは無視されます。-V に続く *security_options* 文字列でキー文字オプションを指定することによって、追加のセキュリティ・サービスを有効化することができます。これらのキー文字は、以下のとおりです。
  - c − データ機密保持サービスを有効にする。
  - i − データ整合性サービスを有効にする。
  - m − 接続の確立に相互認証を有効にする。
  - o − データ・オリジン・スタンピング・サービスを有効にする。
  - r − データ・リプレイの検出を有効にする。
  - q − 順序不整合の検出を有効にする。
- -Z *security_mechanism* は、接続で使用するセキュリティ・メカニズムの名前を指定します。

セキュリティ・メカニズムの名前は、*libtcl.cfg* 設定ファイルで定義されます。*security_mechanism* で名前を指定しない場合は、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。

セキュリティ・メカニズムにログインした後で Adaptive Server にログインする場合は、ユーザ名はセキュリティ・メカニズムから取得されるため、isql -U オプションを指定する必要はありません。次のセッション例を見てみましょう。

```
svrsole4% dce_login user2
Enter Password:
svrsole4% $SYBASE/bin/isql_r -V
1> select suser_name()
2> go

------------------------------
user2
```

この例で、“user2” は dce_login を使用して DCE にログインし、次に -U オプションを指定しないで Adaptive Server にログインします。何もパラメータを指定せずに -V オプションを使用すると、統一化ログインというセキュリティ・サービスを暗黙的に指定することになります。

Adaptive Server の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。

Client-Library を使用して Adaptive Server に接続する場合は、サーバに接続する前にセキュリティ・プロパティを定義できます。たとえば、メッセージの順序をチェックするには、CS_SEC_DETECTSEQ プロパティを設定します。セキュリティ・サービスを Client-Library とともに使用する方法については、『Open Client Client-Library/C リファレンス・マニュアル』を参照してください。

セキュリティ・サービスの使用例

この例では、“mary” というログイン名のユーザが、リモート・プロシージャに DCE セキュリティ・メカニズムを使用するとします。このとき、統一化ログイン (isql または bcp の -V オプションを指定すると必ず有効になる)、メッセージ機密保持、相互認証を使用します。WOND というサーバに接続し、GATEWAY サーバ上で相互認証を使用してリモート・プロシージャを実行します。システム・セキュリティ担当者が WOND と GATEWAY の両方で rpc model B を設定し、このユーザを両方のサーバに追加し、GATEWAY 上でリモートの “trusted” ユーザとして定義していれば、このユーザは以下の手順を実行できます。

1　次のコマンドを使用して DCE セキュリティ・メカニズムにログインし、クレデンシャルを受け取ります。

```
dce_login mary
```

2　isql を使用して Adaptive Server にログインします。

```
isql -SWOND -Vcm
```

3　次のコマンドを実行します。

```
GATEWAY...sp_who
GATEWAY...mary_prc1
GATEWAY...mary_prc2
```

このとき、Mary がサーバとの間で送受信するメッセージはすべて暗号化され (メッセージ機密保持)、リモート・プロシージャを実行するときに、WOND と GATEWAY の両方のサーバが認証されます。

クライアントへのセキュリティ・メカニズムの使用

Adaptive Server は、起動時に、サポートするセキュリティ・メカニズムを決定します。「サポートされているセキュリティ・サービスとメカニズムに関する情報」(494 ページ) を参照してください。Adaptive Server は、サポートするセキュリティ・メカニズムのリストから、特定のクライアントに使用するセキュリティ・メカニズムを選択する必要があります。

クライアントがセキュリティ・メカニズムを指定した場合 (たとえば isql の -Z オプション)、Adaptive Server はそのセキュリティ・メカニズムを使用します。クライアントによる指定がない場合は、*libtcl.cfg* ファイルにリストされている最初のセキュリティ・メカニズムを使用します。

使用できるセキュリティ・サービスの情報の取得

Adaptive Server では、次の情報を取得できます。

- Adaptive Server がサポートしているセキュリティ・メカニズムとセキュリティ・サービス
- 現在のセッションに対してアクティブなセキュリティ・サービス
- 特定のセキュリティ・サービスがセッションに対して有効にされているかどうか

サポートされているセキュリティ・サービスとメカニズムに関する情報

システム・テーブル syssecmechs には、Adaptive Server がサポートしているセキュリティ・メカニズムとセキュリティ・サービスについての情報が格納されています。これは、検索の実行時に動的に作成されるテーブルで、以下のカラムがあります。

- sec_mech_name – セキュリティ・メカニズムの名前。“dce” や “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。“unifiedlogin” など。

このテーブルでは、1 つのセキュリティ・メカニズムに複数のローが存在することがあり、各ローはそのメカニズムでサポートされている個々のセキュリティ・サービスを示します。

Adaptive Server がサポートしているすべてのセキュリティ・メカニズムとセキュリティ・サービスのリストを表示するには、次のクエリを実行します。

```
select * from syssecmechs
```

次のような結果が出力されます。

```
sec_mech_name                  available_service
------------------------------ --------------------
 dce                           unifiedlogin
 dce                           mutualauth
 dce                           delegation
 dce                           integrity
 dce                           confidentiality
 dce                           detectreplay
 dce                           detectseq
```

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

特定のセキュリティ・サービス、たとえば “mutualauth” (相互認証) が有効かどうかを調べるには、is_sec_service_on という関数を使用します。

is_sec_service_on(*security_service_nm*)

security_service_nm は、使用可能なセキュリティ・サービスです。

syssecmechs を問い合わせたときに返されるセキュリティ・サーバを使用します。

たとえば、“mutualauth” (相互認証) が有効かどうかを調べるには、次のコマンドを実行します。

```
select is_sec_service_on("mutualauth")

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

結果が 1 の場合は、このセッションではこのセキュリティ・サービスが有効です。結果が 0 の場合は、このセキュリティ・サービスは使用されていません。

Kerberos の使用

Kerberos は、シークレット・キー暗号法を使用するネットワーク認証プロトコルであり、これによってクライアントがネットワーク接続経由でサーバに ID を証明できます。ユーザがオペレーティング・システムにログインしたとき、または認証プログラムを実行することにより、ユーザ・クレデンシャルが取得されます。このクレデンシャルは、認証を実行するときに各アプリケーションによって使用されます。ユーザは 1 回ログインすれば各アプリケーションにログインする必要はありません。

Kerberos は、KDC (Key Distribution Center) が稼動しており、レルムに対して適切に設定されていることと、クライアント・ライブラリがレルム内の各クライアント・ホストにインストールされていることを前提としています。設定の詳細については、Kerberos のマニュアルと Kerberos ソフトウェアに付属するリファレンス・ページを参照してください。

Adaptive Server では、Kerberos は次のようにサポートされます。

- CyberSafe Kerberos ライブラリ
- MIT Kerberos ライブラリ・バージョン 1.3.1
- ネイティブ・ライブラリ

注意 Kerberos セキュリティ・オプションを有効にするには、「セキュリティ＆ディレクトリサービス」パッケージである ASE_SECDIR が必要です。

Kerberos の互換性

表 16-7 は、各種 Kerberos がサポートされるプラットフォームを示します。

表 *16-7: Adaptive Server* における *Kerberos* の相互運用性

ハードウェア・プラットフォーム	KDC サーバ	GSS (Generic Security Standard) クライアント
Solaris 32	CSF、AD、MIT	CSF、MIT、ネイティブ
Solaris 64	CSF、AD、MIT	CSF、MIT、ネイティブ
Linux 32	CSF、AD、MIT	MIT、ネイティブ
Windows 32	CSF、AD	CSF
AIX 32	CSF	CSF

この相互運用性の表では次の略称を使用しています。

- CSF － CyberSafe 社
- AD － Microsoft Active Directory
- MIT － MIT バージョン 1.3.1

Kerberos 環境での Adaptive Server の起動

Kerberos 環境で Adaptive Server を起動するには、Adaptive Server 名を KDC に追加して、サービス・キーをキー・テーブル・ファイルに抽出します。次に例を示します。

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

注意 管理者は、コマンド・ラインでパスワードを指定する方法で認証を受けることもできます。この例では -k オプションを使用しています。これは、パスワード入力のプロンプトを表示するのではなく、-t オプションで指定した */krb5/v5srvtab* ファイルの中で管理者と Adaptive Server のキーを検索することを管理者に指示するものです。この方法は、シェル・スクリプトを作成する場合に便利です。

Kerberos の設定

設定プロセスは、使用する Kerberos の種類に関係なく共通です。

1　サードパーティ製 Kerberos ソフトウェアを設定して、Kerberos 管理ユーザを作成します。これには、次の処理を行います。

   a　Kerberos クライアント・ソフトウェアを、Open Client Server クライアントまたは Adaptive Server が稼働するマシンにインストールします。次のクライアント・パッケージは動作が確認されています。

      - CyberSafe TrustBroker 4.0
      - MIT Kerberos バージョン 1.3.1

   b　Kerberos KDC サーバを別の専用マシンにインストールします。

   注意 CyberSafe TrustBroker 4.0、MIT Kerberos v.1.3.1、Microsoft Windows Active Directory の KDC は、Adaptive Server とともに使用できることが確認されています。

   c　Kerberos サーバに、管理権限を持つ管理者アカウントを作成します。このアカウントは、後のクライアント作業 (クライアント・マシンでのプリンシパルの作成など) で使用します。

   注意　この後の手順は Kerberos クライアント・マシンで実行します。

2　Adaptive Server の Kerberos プリンシパル *ase120srv* または *ase120srv@MYREALM* を追加します。

3　プリンシパル *ase120srv@MYREALM* の *keytab* ファイルを抽出し、次のようにファイルとして保存します。

```
/krb5/v5srvtab
```

次の UNIX の例では、CyberSafe または MIT Kerberos で利用可能なコマンド・ライン・ツール kadmin を使用します。Kerberos とユーザを管理する GUI ツールもあります。

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

運用環境では、*keytab* ファイルへのアクセスを制御してください。*keytab* ファイルの読み込みを許可されているユーザは、使用しているサーバになり代わるサーバを作成できます。

chmod と chgrp を使用して、*/krb5/v5srvtab* を次のように設定します。

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

Active Directory を KDC として使用するときは、Domain Controller にログインしてユーザと Adaptive Server プリンシパルを追加します。Active Directory ユーザーとコンピュータ・ウィザードを使用して、ユーザとプリンシパルを作成できます。

Adaptive Server で使用する *keytab* ファイルを抽出するには、ktpass というオプション・ツールが必要です。これは、Microsoft サポート ツール・パッケージに含まれています。

Active Directory を使用する場合、ktpass による *keytab* の抽出は、プリンシパルの作成とは別に実行します。Adaptive Server の *keytab* ファイルは、Windows では CyberSafe プログラム・ファイルと同じ場所にあります。たとえば、CyberSafe ソフトウェアが C ドライブにインストールされている場合、Adaptive Server の *keytab* ファイルは *c:¥Program Files¥CyberSafe¥v5srvtab* に格納されると考えられます。

4 ユーザ “sybuser1” の Kerberos プリンシパルを “sybuser1@MYREALM” として追加します。

5　Adaptive Server を起動し、isql を使用して “sa” としてログインします。この後の手順で、Kerberos　セキュリティ・サービスを使用するための Adaptive Server パラメータを設定し、ユーザのログイン・アカウントを作成します。この手順は Windows マシンでも UNIX マシンでも同じです。

- 設定パラメータ use security services を 1 に変更します。

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

- ユーザ “sybuser1” のために新しいログインを追加してから、ユーザを追加します。

```
sp_addlogin sybuser1, password
```

6　Adaptive Server を停止し、管理ファイルと接続設定ファイルを変更します。

- UNIX プラットフォームでは、*$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 プラットフォームでは、*%SYBASE%¥ini* 内に *sql.ini* ファイルがあり、次のように同様のサーバ・エントリが含まれています。

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

- UNIX プラットフォームでは、*$SYBASE/$SYBASE_OCS/config/* に *libtcl.cfg* ファイルまたは *libtcl64.cfg* ファイルがあります。SECURITY セクションに、CyberSafe Kerberos クライアント・ライブラリに関する次のようなエントリが含まれます。

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

  64 ビット版 CyberSafe Kerberos クライアント・ライブラリのエントリは次のようになります。

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

  MIT Kerberos クライアント・ライブラリを使用するマシンでは、エントリは次のようになります。

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

OS 提供のネイティブ・ライブラリを使用するマシン (Linux など) では、エントリは次のようになります。

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

Windows では、*%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かぎりません。

たとえば、CyberSafe UNIX プラットフォームでは、CSFC5KTNAME 環境変数を使用して *keytab* ファイルを指定します。

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

MIT Kerberos でこれに相当する環境変数は KRB5_KTNAME です。

これらの環境変数の詳細については、各ベンダのマニュアルを参照してください。

場合によっては、ダイナミック・ライブラリ検索パスの環境変数を変更する必要があります。UNIX で一般的に使用される環境変数は LD_LIBRARY_PATH です。Windows では通常、PATH が DLL のロケーションを指すように設定されています。アプリケーションでサードパーティのオブジェクトを正しくロードするには、これらの環境変数を変更する必要があります。たとえば、次のコマンドを使用すると、CyberSafe 32 ビット版の *libgss.so* 共有オブジェクトのロケーションが C シェル環境の検索パスに追加されます。

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

8　Adaptive Server を再起動します。次のメッセージが表示されます。

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

9　isql を使用して UNIX ユーザ “sybuser1” として次のように接続します (-U 引数と -P 引数は使用しません)。

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

次のように暗号化オプションを使用することもできます。

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

プリンシパル名の使用

プリンシパル名は、Kerberos KDC (Key Distribution Center) で認証するときにサーバが使用する名前です。複数の Adaptive Server インスタンスを実行中の場合は、Adaptive Server ごとに異なるプリンシパル名を使用する必要があります。

Adaptive Server プリンシパル名の指定

Adaptive Server の名前を指定するには、環境変数 DSLISTEN と DSQUERY、またはコマンドライン・オプション dataserver -s*server_name* を使用します。

プリンシパル名を設定するには、setenv コマンドまたは -k dataserver オプションを使用します。

デフォルトのプリンシパル名は Adaptive Server の名前です。別の名前を指定するには、Adaptive Server を起動して Kerberos を使用する前に、SYBASE_PRINCIPAL を次のように設定します。

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

Adaptive Server のプリンシパル名を設定すると、Adaptive Server はこの変数の値を使用して自身を Kerberos で認証します。

Adaptive Server の起動時に Adaptive Server のプリンシパル名を指定するには、次のコマンドを使用します。

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

Adaptive Server を Kerberos セキュリティ・メカニズムを有効にして起動する場合、Adaptive Server では最初に Kerberos 認証の -k オプションで指定されているプリンシパル名が使用されます。-k オプションが指定されていない場合、Adaptive Server は環境変数 SYBASE_PRINCIPAL でプリンシパル名を確認します。どちらも指定されていない場合、Adaptive Server は認証にサーバ名を使用します。

Adaptive Server では、プリンシパル名のエントリが *keytab* ファイル内に存在する場合に、別のサーバのプリンシパル名を使用する Kerberos Open Client 接続を使用できます。別のプリンシパル名による接続を許可するには、次のいずれかを行います。

- -k オプションのパラメータとして空の文字列を渡す。
- 環境変数 SYBASE_PRINCIPAL を "" に設定する。例 :

```
export SYBASE_PRINCIPAL=""
```

例

この例では、Adaptive Server の名前が “secure_ase”、レルム名は “MYREALM.COM” であり、Adaptive Server の名前は、-s パラメータを使用したコマンド・ラインで dataserver に指定されます。現在のレルムは、secbase 属性値によって *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 Adaptive Server プリンシパル名は “secure_ase@MYREALM.COM” です。Adaptive Server の keytab ファイルで定義されたプリンシパル名が “aseprincipal@MYREALM.COM” の場合、次のオプション 1 または 2 を使用してサーバのプリンシパル名を設定し、デフォルトの Adaptive Server プリンシパル名を上書きできます。

- オプション 1 － -k を指定する。

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

  Kerberos での認証に使用される Adaptive Server のプリンシパル名は “aseprincipal@MYREALM.COM” です。

- オプション 2 － SYBASE_PRINCIPAL を設定する。

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

  Kerberos での認証に使用される Adaptive Server のプリンシパル名は、*$SYBASE_PRINCIPAL* の値の “aseprincipal@MYREALM.COM” です。

- オプション 3 － -k と SYBASE_PRINCIPAL のいずれも設定しない。

  ```
  % $SYBASE/$SYBASE_ASE/bin/dataserver –dmaster.dat
  -s secure_ase
  ```

  Kerberos での認証に使用される Adaptive Server のプリンシパル名は “secure_ase@MYREALM.COM” です。

sybmapname を使用したユーザ・プリンシパル名の処理

sybmapname は、Kerberos 環境で使用される外部のユーザ・プリンシパル名を Adaptive Server のユーザ・ログインのネームスペースに変換します。sybmapname 共有オブジェクトをカスタマイズして、Kerberos 入力バッファで指定された名前を、Adaptive Server 出力バッファへのログインに適した名前にマップできます。

ユーザのプリンシパル名と Adaptive Server のログイン名との間でカスタム・マッピングを実行するには、sybmapname 共有オブジェクトを使用します。この共有オブジェクトは、オプションでサーバの起動時にロードされ、共有オブジェクトに含まれている関数 syb__map_name は、Kerberos 認証が成功した後およびユーザ・プリンシパルが syslogins テーブル内のログインにマップされる直前に呼び出されます。この関数は、マップされるユーザのプリンシパル名とログイン名が同一ではない場合に役に立ちます。

```
syb__map_name(NAMEMAPTYPE *protocol, char *orig,
int origlen, char *mapped, int *mappedlen)
```

各パラメータの意味は、次のとおりです。

- `NAMEMAPTYPE *protocol` － この関数の使用のために予約されている構造体を表す。
- `char *orig` － Null で終了しない入力バッファ。
- `int origlen` － 入力バッファの長さ。255 文字以内にする必要がある。
- `char *mapped` － Null で終了しない出力バッファ。
- `int *mappedlen` － 出力バッファの長さ。30 文字以内にする必要がある。

syb__map_name は、マッピングが成功した場合は 0 よりも大きい値を返し、マッピングが実行されなかった場合は 0 の値を返し、syb__map_name でエラーが発生した場合は 0 よりも小さい値を返します。エラーが発生すると、Adaptive Server のエラー・ログにマッピングの失敗をレポートするメッセージが書き込まれます。

たとえば、Adaptive Server で Kerberos ユーザを認証するには、次の手順に従います。

1　Kerberos セキュリティ・メカニズムを使用するように Adaptive Server を設定します。「Kerberos の使用」(495 ページ) と Open Client/Server マニュアル、および Sybase Web サイト (http://www.sybase.com/detail?id=1029260) のホワイト・ペーパー「Configuring Kerberos for Sybase」を参照してください。

   サンプルの *sybmapname.c* ファイルは、*$SYBASE/$SYBASE_ASE/sample/server/sybmapname.c* にあります。

2　*sybmapname.c* を修正して、ロジックを実装します。「sybmapname を使用する場合の注意事項」(506 ページ) を参照してください。

3 提供されている汎用プラットフォーム固有の makefile を使用して共有オブジェクトまたは DLL を構築します。makefile は、プラットフォーム固有の設定に合わせて変更しなければならない場合があります。

4 生成された共有オブジェクトは、UNIX マシンでは $LD_LIBRARY_PATH で指定したロケーション、Windows マシンでは PATH 変数で指定したロケーションに保存されます。ファイルには、“sybase” ユーザに対する読み取りおよび実行パーミッションが必要です。

注意 “sybase” ユーザにのみ読み取りパーミッションや実行パーミッションを許可し、他のアクセスはすべて拒否することをおすすめします。

Kerberos 認証を使用した Adaptive Server へのログインの確認

Kerberos 認証を使用して Adaptive Server へのログインを確認するには、次のことを前提とします。

- *$SYBASE* は、リリースおよびインストールのディレクトリを参照する。
- *$SYBASE_ASE* は、サーバ・バイナリを含む Adaptive Server バージョン・ディレクトリを参照する。
- *$SYBASE_OCS* は、Open Client/Server バージョン・ディレクトリを参照する。

例 1 クライアントのプリンシパル名が `user@REALM` であり、syslogins テーブル内の対応するエントリが `user_REALM` である場合は、入力文字列 `user@realm` を受け取って、その入力文字列を出力文字列 `user_REALM` に変換するように sybmapname をコード化できます。

例 2 クライアントのプリンシパル名が `user` であり、syslogins テーブル内の対応するエントリが `USER` である場合は、入力文字列 `user` を受け取って、この文字列を大文字の文字列 `USER` に変換するように sybmapname をコード化できます。

sybmapname は、Adaptive Server によって実行時に読み込まれ、そのロジックを使用して必要なマッピングを実行します。

次の操作と出力は、例 2 で説明する sybmapname 関数を示しています。syb__map_name()に対してカスタマイズされた定義を含む *sybmapname.c* ファイルはコンパイルして、共有オブジェクト (または DLL) としてビルドした後に、適切なパスのロケーションに保存する必要があります。Kerberos のセキュリティ・メカニズムを有効にして Adaptive Server を起動します。

TGT (Ticket Granted Ticket) は、識別情報を提供する、暗号化形式のファイルです。このファイルを初期化するには、次のように入力します。

```
$ /krb5/bin/kinit johnd@public
Password for johnd@public:
$
```

TGT を一覧表示するには、次のように入力します。

```
$ /krb5/bin/klist
   Cache Type:Kerberos V5 credentials cache
   Cache Name:/krb5/tmp/cc/krb5cc_9781
Default principal:johnd@public
```

“sa” としてログインし、“johnd” のユーザ・ログインを確認します。

```
$ $SYBASE/$SYBASE_OCS/bin/isql -Usa -P
      -Ipwd`/interfaces
1>

1> sp_displaylogin johnd
2> go
No login with the specified name exists.
(return status = 1)

1> sp_displaylogin JOHND
2> go
Suid: 4
Loginame:JOHND
Fullname:
Default Database:master
Default Language:
Auto Login Script:
Configured Authorization:
Locked:NO
Password expiration interval: 0
Password expired:NO
Minimum password length: 6
Maximum failed logins: 0
Current failed login attempts:
Authenticate with:ANY
(return status = 0)
```

Kerberos 認証が成功すると、sybmapname ユーティリティを使用して小文字の johnd が大文字の JOHND にマップされ、ユーザ johnd は Adaptive Server にログインできるようになります。

```
$ $SYBASE/$SYBASE_OCS/bin/isql -V -I'pwd'/interfaces
1>
```

sybmapname を使用する場合の注意事項

sybmapname のコーディングを行う場合は、次の点に注意する必要があります。

- サンプルの *sybmapname.c* プログラムに変更を加える場合は、慎重に行う必要があります。セグメンテーション・フォールトを発生させるコード、exit を呼び出すコード、system calls を呼び出すコード、UNIX シグナルを変更するコード、ブロック呼び出しを行うコードの使用は避けてください。不適切なコーディングや呼び出しは、Adaptive Server エンジンを妨害する場合があります。

  注意 sybmapname におけるコード・エラーは、Sybase の責任ではありません。

- コードを注意深く作成し、すべてのポインタをチェックしてから参照を解除して、システム・コールを回避します。記述する関数は、クイック・ネーム・フィルタリング関数にする必要があります。
- goto 文を使用しないでください。プラットフォームによっては、これらの文によって予期しない悪影響を受ける場合があります。
- 複数のレルムを使用する場合は、ユーザ・プリンシパル名を適切なログイン名に注意深くマップし、レルム情報が反映されるようにします。たとえば、ユーザ・プリンシパル名 `userA@REALMONE` と `userB@REALMTWO` をそれぞれ持つ 2 人のユーザがいる場合、ログイン名 `userA_REALMONE` と `userB_REALMTWO` にマップします。`userA` または `userB` にはマップしないでください。この動作により、異なるレルムに属する 2 人のユーザが区別されます。

Kerberos による同時認証

以前のバージョンでは、Kerberos による認証時にロック・メカニズムを使用することによって内部データ構造を保護していましたが、Adaptive Server バージョン 15.0.3 では、Kerberos による同時認証がサポートされるようになりました。

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

バージョン 15.0.3 では、Kerberos による認証時に同時ログイン・セッションがブロックされる問題も解決されています。同時実行性に関連したこの問題は、以前のバージョンの Adaptive Server を、MIT バージョン 1.3.x および 1.4.x の Kerberos GSSAPI ライブラリとともに使用する場合に発生します。

LDAP ユーザ認証のための Adaptive Server の設定

LDAP ユーザ認証を使用すると、クライアント・アプリケーションは Adaptive Server にユーザ名とパスワードの情報を送信し、syslogins ではなく LDAP サーバによる認証を行えるようになります。LDAP サーバを使用する認証では、Adaptive Server またはアプリケーション固有のパスワードではなく、サーバ全体のパスワードを使用できます。

LDAP ユーザ認証は、ユーザ管理を単純化して集中化する場合や、ユーザ管理が煩雑にならないようにする場合に最適な方法です。

LDAP ユーザ認証は、LDAP プロトコル標準バージョン 3 に準拠したディレクトリ・サーバ (Active Directory、iPlanet、OpenLDAP Directory Server など) で動作します。

LDAP ユーザ認証では、次のいずれかの認証アルゴリズムを使用します。

- 生成 DN (認証用、Adaptive Server バージョン 12.5.1 以降で使用可能)
- 検索 DN (Adaptive Server バージョン 12.5.2 以降で使用可能)

各アルゴリズムは、ユーザの DN (識別名) を取得する方法が異なります。

LDAP プロトコルで使用されるプライマリ・データ構造は LDAP URL です。

LDAP URL は、LDAP サーバ上のオブジェクトまたは値のセットを指定します。Adaptive Server は、LDAP URL を使用して、ログイン要求の認証に使用する LDAP サーバと検索基準を指定します。

LDAP URL では、次の構文を使用します。

```
ldapurl::=ldap://host:port/node/attributes [base | one | sub] filter
```

各パラメータの意味は、次のとおりです。

- *host* － LDAP サーバのホスト名。
- *port* － LDAP サーバのポート番号。
- *node* － 検索を開始するオブジェクト階層内でのノードを指定する。
- *attributes* － 結果セットで返す属性のリスト。属性リストは、LDAP サーバによって異なることがある。
- base | one | sub － 検索条件を修飾する。base は、ベース・ノードの検索を指定する。one は、node で指定されたベース・ノードとその 1 つ下のレベルのノードの検索を指定する。sub は、node で指定されたベース・ノードとその下位レベルのすべてのノードの検索を指定する。
- filter － 認証する属性を指定する。フィルタは、`uid=*` のように簡潔にすることも、`(uid=*)(ou=group)` のように複雑にすることもできる。

生成 DN アルゴリズム

生成 DN アルゴリズムを使用する場合、ログインは次の手順で行われます。

1 Open Client は、Adaptive Server のリスナ・ポートに接続します。

2 Adaptive Server リスナは、接続を受け付けます。

3 Open Client は、内部ログイン・レコードを送信します。

4 Adaptive Server は、ログイン・レコードを読み込みます。

5 Adaptive Server は、プライマリ URL から生成した DN とログイン・レコードのログイン名を使用して LDAP サーバにバインドします。このとき、ログイン・レコードのパスワードも使用します。

6 LDAP サーバは、ユーザを認証し、成功か失敗かを示すメッセージを返します。

7 プライマリ URL で検索が指定されている場合、Adaptive Server は LDAP サーバに検索要求を送信します。

8 LDAP サーバは、検索結果を返します。

9 Adaptive Server は、検索結果に基づいてログインを受け付けるか、または拒否します。

検索 DN アルゴリズム

検索 DN アルゴリズムを使用する場合、ログインは次の手順で行われます。

1 Open Client は、Adaptive Server のリスナ・ポートに接続します。

2 Adaptive Server リスナは、接続を受け付けます。

3 Open Client は、内部ログイン・レコードを送信します。

4 Adaptive Server は、ログイン・レコードを読み込みます。

5 Adaptive Server は、ディレクトリ・サーバのアクセス・アカウントを使用して LDAP サーバにバインドします。

   手順 5 ～ 6 で確立された接続は、次に Adaptive Server が認証を試行して DN 検索への接続を再利用するまで継続します。

6 LDAP サーバは、ユーザを認証し、成功か失敗かを示すメッセージを返します。

7 Adaptive Server は、ログイン・レコードのログイン名と DN 検索 URL に基づいて、LDAP サーバに検索要求を送信します。

8 LDAP サーバは、検索結果を返します。

9 Adaptive Server は、検索結果を読み込み、DN 検索 URL から属性値を取得します。

10　Adaptive Server は、取得した属性値を DN として使用し、パスワードを使用して LDAP サーバにバインドします。

11　LDAP サーバは、ユーザを認証し、成功か失敗かを示すメッセージを返します。

12　プライマリ URL で検索が指定されている場合、Adaptive Server は LDAP サーバに検索要求を送信します。

13　LDAP サーバは、検索結果を返します。

14　Adaptive Server は、検索結果に基づいてログインを受け付けるか、または拒否します。

上記のいずれかの認証基準が満たされない場合、Adaptive Server は一般的なログインの失敗をレポートします。

プライマリ URL 文字列またはセカンダリ URL 文字列の検索基準を指定しない場合は、手順 12 ～ 13 を省略できます。認証が完了し、手順 11 で返される成功か失敗かを示すメッセージが表示されます。

LDAP の設定

Adaptive Server で LDAP ユーザ認証を設定するには、次の手順を実行します。

新しい Adaptive Server での LDAP の設定

1　Adaptive Server の LDAP URL 検索文字列とアクセス・アカウントの値を指定します。

2　enable ldap user auth を 2 に設定します。

3　LDAP ベンダ提供のツールを使用して、LDAP ディレクトリ・サーバにユーザを追加します。

4　sp_addlogin を使用して、Adaptive Server にユーザを追加します。また、sp_maplogin を使用すると、認証時にログイン・アカウントが自動的に作成されるように設定したり、他のログイン制御を適用したりできます。

既存の Adaptive Server の LDAP へのマイグレーション

既存のサーバでサービスが中断されないようにするには、次の操作を実行して Adaptive Server を LDAP にマイグレートします。

- Adaptive Server に LDAP URL 検索文字列を指定します。
- 構成パラメータ enable ldap user auth を 1 に設定します。
- LDAP ディレクトリ・サーバにユーザを追加します。
- すべてのユーザを LDAP サーバに追加するときに、すべての認証が LDAP で行われるようにするには、enable ldap user auth を 2 に設定するか、sp_maplogin を使用してログイン制御で設定パラメータを上書きします。

LDAP ユーザ認証の管理

LDAP URL 検索文字列の作成または表示、LDAP URL 検索文字列またはログインの確認、アクセス・アカウントとチューニング可能な LDAPUA (LDAP ユーザ認証) 関連パラメータの指定には、sp_ldapadmin を使用します。sp_ldapadmin を実行するには、システム・セキュリティ担当者 (SSO) の役割が必要です。

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

生成 DN アルゴリズムの例

使用する LDAP サーバのトポロジとスキーマが単純な場合は、ユーザ認証に生成 DN アルゴリズムを使用できます。商用のスキーマ (iPlanet ディレクトリ・サーバや OpenLDAP ディレクトリ・サーバなど) を使用する場合、ユーザは LDAP サーバ・ツリー内の同じコンテナ内のオブジェクトとして作成され、このオブジェクトのロケーションに基づいてユーザの DN が決定されます。ただし、LDAP サーバのスキーマには以下の制限があります。

- 認証されるユーザをユニークに識別する属性名を含むフィルタを指定する必要がある。
- 属性 `name=*` を含むフィルタを指定する必要がある。アスタリスクはワイルドカード文字。フィルタに使用する属性名は、LDAP サーバのスキーマによって異なる。
- Adaptive Server のログイン名は、UNIX ユーザ名などと同様の短縮ユーザ名である。
- DN は、埋め込みスペースや句読表記を含むフル・ネームではなく、短縮ユーザ名を使用する。たとえば、jqpublic は DN の制限事項を満たしているが、“John Q. Public” は満たしていない。

iPlanet の例

LDAP のベンダによっては、以下の例で使用している以外のオブジェクト名、スキーマ、属性を使用することがあります。使用できる LDAP URL 検索文字列は数多くあります。また、有効なサイトがスキーマをローカルに拡張したり、サイトごとに異なる方法でスキーマを使用したりすることもできます。

- 次の例では、`uid=*` フィルタを使用しています。Adaptive Server は、このフィルタのワイルドカードを認証対象となる Adaptive Server のログイン名に置換してから、LDAP URL のノード・パラメータに追加して DN を生成します。生成される DN は次のとおりです。

  ```
  uid=myloginname,ou=People,dc=mycomany,dc=com
  ```

- Adaptive Server は、バインド操作に成功した後、接続を使用して `uid` などの属性名を検索します。この属性名は、ログイン名と同じです。

```
sp_ldapadmin set_primary_url,
'ldap://myhost:389/ou=People,dc=mycompany,dc=com??sub?uid=*'
```

- 次の例では、OpenLDAP 2.0.25 で定義された、属性名 cn を含むスキーマを使用しています。

  生成 DN は cn=myloginname,dc=mycompany,dc=com です。

  ```
  sp_ldapadmin set_primary_url,
  'ldap://myhost:389/dc=mycompany,dc=com??sub?cn=*'
  ```

検索 DN アルゴリズムの例

生成 DN アルゴリズムを使用するための制限事項を満たしていない Active Directory サーバまたはその他の LDAP サーバ環境を使用する場合は、検索 DN アルゴリズムを使用します。

- Windows 2000 Server で提供されている商用のユーザ・スキーマを使用する Active Directory サーバの場合は、以下の手順を実行します。

  a　アクセス・アカウント情報を設定します。

```
sp_ldapadmin set_access_acct,
'cn=Admin Account, cn=Users, dc=mycompany, dc=com',
'Admin Account secret password'
```

  b　プライマリ URL を設定します。

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

  c　DN 検索 URL 検索文字列を設定します。

```
sp_ldapadmin set_dn_lookup_url,
'ldap://hostname:389/cn=Users,dc=mycompany,dc=com?distinguishedName?one?sama
ccountname=*'
```

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。この属性名を使用して、Adaptive Server のログイン名が検索されます。ユーザの DN には、句読表記と埋め込みスペースを含むフル・ネーム (たとえば、cn=John Q. Public, cn=Users, dc=mycomany, dc=com) が使用されます。Windows の DN では短縮名を使用しないため、検索 DN アルゴリズムは、LDAP サーバに Active Directory スキーマ (デフォルト) を使用しているサイトに適しています。プライマリ URL は検索を指定しません。代わりに、バインド操作を使用して認証を行います。

検索フィルタによる Adaptive Server へのアクセスの制限例

LDAP URL 検索文字列を使用して、LDAP サーバ上の特定のユーザ・グループだけにアクセスを制限できます。たとえば、accounting グループのユーザだけがログインできるようにするには、複合フィルタを使用して、属性が group=accounting のユーザのグループだけにアクセスを制限します。

- 次の LDAP URL 文字列では、iPlanet サーバに生成 DN アルゴリズムを使用しています。

  ```
  sp_ldapadmin set_primary_url,
  'ldap://myhost:389/ou=People,dc=mycompany,
  dc=com??sub?(&(uid=*)(group=accounting))'
  ```

Adaptive Server は、`uid=mylogin,ou=People,dc=mycompany,dc=com` という DN を使用してバインドします。この ID を使用したバインドが成功すると、Adaptive Server は次のように検索します。

```
"ou=People,dc=mycompany,dc=com??sub?(&(uid=mylogin)(group=accounting))"
```

この検索からオブジェクトが返されると、認証が成功します。

- 以下の例では、LDAP URL 検索文字列と複合フィルタを使用しています。

```
sp_ldapadmin set_primary_url,
'ldap://myhost:389/ou=people,dc=mycompany,dc=com??sub?(&(
uid=*)(ou=accounting) (l=Santa Clara))'

sp_ldapadmin, set_primary_url,
'ldap://myhost:389/ou=people,dc=mycompany,dc=com??sub?(&(
uid=*)(ou=Human%20Resources))'
```

LDAP ユーザ認証パスワード情報の変更

Adaptive Server が LDAP サーバから取得してクライアントに渡す、LDAP ユーザ認証関連の通知メッセージが 2 つあります。

- 期限が切れそうな LDAP ユーザ認証パスワードを使用する LDAP 認証メカニズムを使用して Adaptive Server にログインした場合は、次のメッセージが表示される。

  パスワードはあと *<number>* 日で有効期限が切れます。

- LDAP サーバ管理者がパスワードをリセットした後、または LDAP サーバのパスワードの期限が切れた後に、LDAP 認証メカニズムを使用して Adaptive Server にログインすると、次のメッセージ 4002 が表示される。

  ログインに失敗しました

  次のように、監査が有効で、errors 監査オプションがオンになっている場合は、メッセージ 4099 が監査ログに送信される。

  LDAP パスワードの期限が切れました。

注意 この追加の情報を提供できるように LDAP サーバを設定してください。また、Adaptive Server は、LDAP クライアントに対する LDAP パスワード制御の転送をサポートしている必要があります。

フェールオーバのサポート

プライマリ URL で指定された LDAP ディレクトリ・サーバで重大な障害が発生し、ネットワーク要求に応答しなくなった場合、Adaptive Server はセカンダリ URL で指定されたセカンダリ LDAP ディレクトリ・サーバに接続しようとします。Adaptive Server は、LDAP 関数 ldap_init を使用して、LDAP ディレクトリ・サーバへの接続をオープンできるかどうかを調べます。プライマリ URL 文字列が NULL または無効である場合、Adaptive Server はセカンダリ URL へのフェールオーバを試行します。LDAP のバインド操作や検索操作で障害が発生した場合、Adaptive Server はセカンダリ URL にフェールオーバしません。

Adaptive Server ログインと LDAP ユーザ・アカウント

LDAP ユーザ認証を有効にし、認証アルゴリズムと URL 文字列の選択と設定を行ったら、ユーザ・アカウントを設定します。LDAP 管理者が LDAP サーバのアカウントの作成と管理を行い、データベース管理者が Adaptive Server のアカウントの作成と管理を行います。また、データベース管理者は、管理オプションを使用して、Adaptive Server と LDAP サーバなどの外部認証メカニズムを統合するときのログイン・アカウントを柔軟に設定できます。データベース管理者は、従来のコマンドとプロシージャを使用して、Adaptive Server アカウントの役割、デフォルト・データベース、デフォルト言語、およびその他のログイン固有の属性の管理を続行できます。

表 16-8 は、ログイン時の syslogins テーブルの変更を示します。ここに示す変更は、LDAP ユーザ認証が設定済みで、ログインが LDAP の使用を制限されておらず、create login マッピングを設定していないことを前提としています。

表 *16-8: LDAP* による *syslogins* の変更

syslogins にそのユーザのローが既に存在する	LDAP サーバ認証に成功	syslogins の変更
いいえ	はい	変更なし、ログインは失敗
いいえ	いいえ	変更なし、ログインは失敗
はい	はい	パスワードが変更された場合は、ローが更新される
はい	いいえ	変更なし

セカンダリ検索サーバのサポート

Adaptive Server では、LDAP サーバによって認証された Adaptive Server クライアントの継続的なサポートが提供されます。LDAP サーバで障害が発生した場合や、計画されたダウンタイムがある場合に、プライマリ LDAP サーバからフェールオーバするセカンダリ LDAP 検索サーバを指定できます。

URL セットの状態は、次のステータスを通じて監視されます。

- INITIAL − LDAP ユーザ認証が設定されていないことを示す。
- RESET − Adaptive Server の管理コマンドで URL が入力されていることを示す。
- READY − URL が接続を受け入れる準備ができていることを示す。
- ACTIVE − URL で LDAP ユーザ認証が成功したことを示す。
- FAILED − LDAP サーバへの接続中に問題が発生したことを示す。
- SUSPENDED − URL がメンテナンス・モードになっており、使用されないことを示す。

次の手順で、フェールオーバと手動によるフェールバックについて説明します。

1 プライマリおよびセカンダリの URL セットが設定されて READY ステータスになります。

2 接続が、プライマリ・サーバ・インフラストラクチャを使用して認証されます。

3 プライマリ・サーバで障害が発生すると、ステータスが FAILED に変わります。

4 セカンダリ・サーバ・インフラストラクチャによる認証が接続で自動的に開始されます。

5 LDAP 管理者によってプライマリ・サーバが修復されて、オンラインに戻ります。Adaptive Server 管理者によりプライマリ LDAP サーバのステータスが READY に変更されます。

6 新しい接続が、プライマリ・サーバを使用して認証されます。

注意 Adaptive Server がセカンダリ LDAP サーバにフェールオーバしたら、データベース管理者は、プライマリ LDAP サーバを手動でアクティブにしてから使用する必要があります。

Adaptive Server で LDAP サーバへの接続時にエラーが発生した場合は、認証が 3 回再試行されます。エラーが続く場合、LDAP サーバのステータスは FAILED になります。Adaptive Server で再試行ループが発生する原因となる LDAP エラーについては、「LDAP ユーザ認証エラーのトラブルシューティング」(521 ページ) を参照してください。

セカンダリ検索 LDAP サーバを設定するには、sp_ldapadmin を使用します。

- セカンダリ DN 検索 URL を設定するには、次のように入力します。

  ```
  sp_ldapadmin set_secondary_dn_lookup_url, <URL>
  ```

- セカンダリ DN 検索 URL の管理アクセス・アカウントを設定するには、次のように入力します。

```
sp_ldapadmin set_secondary_access_acct, <DN>, <password>
```

- 認証のためにプライマリまたはセカンダリ URL の使用をサスペンドするには、次のように入力します。

  ```
  sp_ldapadmin suspend, {primary | secondary}
  ```

- 認証のためにプライマリまたはセカンダリ URL のセットをアクティブ化するには、次のように入力します。

  ```
  sp_ldapadmin activate, {primary | secondary}
  ```

- プライマリおよびセカンダリ LDAP サーバの設定およびステータスの詳細を表示するには、次のように入力します。

  ```
  sp_ldapadmin list
  ```

  sp_ldapadmin list は、list_access_acct および list_urls からの前回の出力を結合します。プライマリ・サーバとセカンダリ・サーバでは次が出力されます。

  - 検索 URL
  - 識別名検索 URL
  - アクセス・アカウント DN
  - アクティブ [true | false]
  - ステータス [ready | active | failed | suspended | reset]

Adaptive Server バージョン 12.5.4 以降には、セカンダリ・サーバをサポートする、次の sp_ldapadmin オプションが用意されています。

- セカンダリ・サーバの DN 検索 URL を表示するには、次のように入力します。

  ```
  sp_ldapadmin list_urls
  ```

- セカンダリ DN 検索の管理アクセス・アカウントを表示するには、次のように入力します。

  ```
  sp_ldapadmin list_access_acct
  ```

- サブコマンドを表示するには、次のように入力します。

  ```
  sp_ldapadmin help
  ```

LDAP サーバのステータスの移行

表 16-9 ～表 16-14 に、sp_ldapadmin の各コマンドを実行したときの LDAP サーバのステータスの移行を示します。

表 16-9 に、sp_ldapadmin set_URL を実行したときのステータスの移行を示します。ここで set_URL は次のコマンドのいずれかを表します。

- set_dn_lookup_url
- set_primary_url
- set_secondary_dn_lookup_url
- set_secondary_url

表 *16-9: sp_ldapadmin set_URL* 実行時のステータスの移行

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

表 16-10 に、sp_ldapadmin suspend を実行したときのステータスの移行を示します。

表 *16-10: 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 16-11 に、sp_ldapadmin activate を実行したときのステータスの移行を示します。

表 *16-11: sp_ldapadmin set activate* 実行時のステータスの移行

初期状態	最終状態
INITIAL	エラー
RESET	READY
READY	READY
ACTIVE	ACTIVE
FAILED	READY
SUSPENDED	READY

次の表に、Adaptive Server で暗黙に実行される LDAP サーバのステータスの移行を示します。

表 16-12 に、Adaptive Server を再起動するときのステータスの移行を示します。

表 *16-12: Adaptive Server* 再起動時のステータスの移行

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

Adaptive Server は、LDAP サーバが READY または ACTIVE ステータスの場合にのみ LDAP ログインを実行します。表 16-13 は、ステータスの移行を示します。

表 *16-13: 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 16-14 に LDAP ログインが失敗した場合のステータスの移行を示します。

表 *16-14: LDAP* ログイン失敗時のステータスの移行

初期状態	最終状態
READY	FAILED
ACTIVE	FAILED

LDAP ユーザ認証のチューニング

Adaptive Server のオプションの設定とチューニングは、着信接続の負荷および Adaptive Server-LDAP サーバ・インフラストラクチャに基づいて行います。同時着信要求の数に基づいて、以下のオプションを設定します。

- sp_configure を使用して、エンジンあたりのネイティブ・スレッド数を指定する max native threads を設定する。
- sp_ldapadmin を使用して、エンジンあたりの LDAP ユーザ認証ネイティブ・スレッド数を指定する max_ldapua_native_threads を設定する。

ネットワークおよび Adaptive Server/LDAP サーバ・インフラストラクチャの状態に基づいて、(LDAP サーバのバインドおよび検索タイムアウトを指定する) set_timeout オプションを設定します。

set_abandon_ldapua_when_full オプションを設定して、着信接続が max_ldapua_native_threads に達した場合の Adaptive Server の動作を指定します。

パフォーマンスを向上させるように LDAP サーバを設定するには、以下の sp_ldapadmin オプションを使用します。

- set_max_ldapua_desc － LDAPUA 接続要求の同時実行性を管理します。識別名アルゴリズムを使用している場合に、set_max_ldapua_desc の値を大きくすると、Adaptive Server による LDAPUA 接続の処理が高速化します。
- set_num_retries － 試行回数を設定します。この値は、Adaptive Server と LDAP サーバとの間の一時的なエラーの数に基づいて調整します。再試行回数を設定すると、一時的なエラーを取り消すことができます。
- set_log_interval － Adaptive Server のエラー・ログに診断目的で送信されるメッセージの数を制御します。小さい値を指定すると、エラー・ログがメッセージで混雑しますが、特定のエラーを調べるときに効果的です。大きい値を指定すると、エラー・ログに送信されるメッセージの数が少なくなりますが、エラーを調べるための効果は低くなります。set_log_interval は、エラー・ログのサイズに合わせて調整します。

ログイン・マッピングに対する制御の強化

sp_maplogin を使用して、LDAP または PAM で認証されるユーザをローカルの Adaptive Server ログインにマップします。

注意 Kerberos で認証されたユーザをマップするには、sp_maplogin ではなく、sybmapname を使用します。

---

sp_maplogin を使用してログイン・マッピングを作成または変更できるのは、sso_role を持っているユーザだけです。

Adaptive Server では、ログインの認証メカニズム設定とログインを使用するマッピング間の競合が回避されます。潜在的なマッピングの競合は、ストアド・プロシージャ sp_maplogin、sp_modifylogin、または sp_addlogin によって検出されます。

これらのコントロールでは、以下のマッピングは許可されていません。

- 1 つの Adaptive Server ログイン名から別のログイン名へのマッピング
- ローカルのログインとして既に存在している外部名からのマッピング
- 存在しないログイン名へのマッピング

また、マッピングを使用して認証メカニズムが指定されている場合、メカニズムはターゲットのログインに設定されている認証メカニズムによりチェックされます。

ターゲットのログインの認証メカニズムによって、特定の認証メカニズムを使用するようにログインが制限されている場合は、マッピングで指定されたメカニズムはログインに指定されているメカニズムに一致するか、“ANY” 認証メカニズムと一致する必要があります。

sp_maplogin で、競合が存在することが検出されると、sp_maplogin は失敗し、競合を特定するエラーがレポートされます。

同様に、sp_modifylogin および sp_addlogin 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。sp_modifylogin または sp_addlogin で競合が検出されると、ログイン・マッピングとの競合を特定するためのエラーがレポートされます。

例

例 1　LDAP ユーザを Adaptive Server の “sa” ログインにマップします。ある企業は、すべてのユーザ・アカウントに対するレポジトリとして LDAP を採用しており、数百台の Adaptive Server を管理できるデータベース管理者 “adminA” および “adminB” を含むすべてのユーザに LDAP 認証を要求するセキュリティ・ポリシーを使用しています。監査は有効になっており、ログイン・イベントは、監査証跡に記録されます。

これらの管理アカウントを “sa” にマップするには、次のように入力します。

```
sp_maplogin LDAP, 'adminA', 'sa'
go
sp_maplogin LDAP, 'adminB', 'sa'
go
```

次のように入力して、LDAP 認証を使用した認証をすべてのユーザに対して要求します。

```
sp_configure 'enable ldap user auth', 2
go
```

“adminA” が Adaptive Server へのログイン中に認証されると、“sa” だけでなく “adminA” に関連付けられた識別名がログイン監査イベントに記録されます。これにより、アクションを実行している各ユーザを監査証跡で識別することができます。

“adminA” および “adminB” のパスワードが LDAP サーバで設定されている場合は、管理対象のすべての Adaptive Server で “sa” パスワードを維持する必要はありません。

この例では、外部の異なる ID やパスワードを認証に使用することもできますが、Adaptive Server 内でこれを行うには、“sa” アカウントに関連付けられた特殊な権限も必要です。

例 2 PAM および LDAP の両方を使用してアプリケーション・ログインにユーザをマップします。ある企業は、PAM および LDAP 認証の両方を採用していますが、それぞれ別の目的で使用しています。会社のセキュリティ・ポリシーでは、LDAP を一般的なユーザ・アカウントの認証メカニズムとして定義し、PAM を中間層アプリケーションなどの特殊なユーザ用として定義しています。中間層アプリケーションは、Adaptive Server への接続プールを設定して、中間層アプリケーションのユーザに代わって要求を処理する場合があります。

LDAP および PAM 両方のユーザ認証のための Adaptive Server の設定は、次のように行います。

```
sp_configure 'enable ldap user auth', 2
go
sp_configure 'enable pam user auth', 2
go
```

Adaptive Server のログイン `appX` を、中間層アプリケーションに適したパーミッションを使用してローカルに設定します。

```
sp_addlogin 'appX', password
go
sp_modifylogin appX, 'authenticate with', PAM
go
```

単純なパスワードを “appX” にハードコードしていくつかの異なる Adaptive Server でそのパスワードを統一して管理するのではなく、中間層アプリケーションを検証するための追加の情報を使用して中央レポジトリでアプリケーションを認証するカスタムの PAM モジュールを開発します。

クライアント・アプリケーションのログイン “appY” には、LDAP ID とパスワードによるユーザの LDAP 認証が必要です。すべての LDAP 認証ユーザをログイン “appY” にマップするには、sp_maplogin を使用します。

```
sp_addlogin 'appY', password
go
sp_maplogin LDAP, NULL, 'appY'
go
```

“appY” のユーザは会社の ID とパスワードを使用して認証されてから、ローカルの Adaptive Server のログイン “appY” にマップされ、データベース・アクションを実行します。LDAP ユーザの ID を使用して認証が行われると、監査証跡に記録され、アプリケーションのログイン “appY” に適したパーミッションで実行されます。

LDAP ユーザ認証エラーのトラブルシューティング

Adaptive Server では、LDAP サーバと通信中に次のような一時的なエラーが発生する場合があります。通常、接続を再試行するとこれらのエラーは解決します。再接続した後も同様のエラーが解決しない場合は、Adaptive Server によって LDAP サーバに FAILED のステータスが設定されます。

- LDAP_BUSY − サーバがビジー。
- LDAP_CONNECT_ERROR − 接続中のエラー。
- LDAP_LOCAL_ERROR − クライアント側のエラー。
- LDAP_NO_MEMORY − クライアント側にメモリを割り付けることができない。
- LDAP_OPERATIONS_ERROR − サーバ側のエラー。
- LDAP_OTHER − 不明なエラー・コード。
- LDAP_ADMINLIMIT_EXCEEDED − 検索が制限を超えている。
- LDAP_UNAVAILABLE − サーバが要求を処理できない。
- LDAP_UNWILLING_TO_PERFORM − サーバが要求を処理しない。
- LDAP_LOOP_DETECT − 参照中にループが検出された。
- LDAP_SERVER_DOWN − サーバに到達できない (接続が失敗した)。
- LDAP_TIMEOUT − ユーザ指定の時間内にオペレーションが完了しないために LDAP API が失敗した。

一時的なエラーや多数の同時ログイン要求によって、エラー・ログで大量のエラー・メッセージが繰り返される場合があります。ログを読みやすくするために、次のエラー・メッセージ・ログ・アルゴリズムが使用されます。

1　初めてログに記録されるメッセージは、そのまま記録されます。

2　メッセージが最後に記録されてから 3 分を超えた場合は、次のようになります。

   - エラー・メッセージが記録される。
   - メッセージが最後に出力されてからメッセージが繰り返された回数が記録される。
   - メッセージが出力されてから経過した時間が分単位で記録される。

次の原因で発生した認証エラーは、LDAP エラーとは見なされず、認証要求を再試行する条件にはなりません。

- 不正なパスワードまたは無効な識別名によるバインド・エラー。
- 0 の結果セットを返すか、属性値を返さない、バインドが成功した後の検索。

URL 解析中に検出される構文エラーは LDAP URL の設定時にキャッチされるため、上記のいずれのカテゴリにも該当しません。

LDAP サーバの設定

LDAP (Lightweight Directory Access Protocol) のユーザ認証では、SSL/TLS (Secure Sockets Layer/Transport Layer Security) プロトコルがサポートされており、Adaptive Server と LDAP サーバ間でのデータ転送の安全性を確保できます。

❖ LDAP サーバへの接続の設定

1 信頼されたルート証明書がすべて同じファイルに保存されていることを確認します。

信頼されたサーバを定義すると、Adaptive Server によってセキュア接続が次のように設定されます。ここで、*servername* は現在の Adaptive Server の名前です。

- *$SYBASE_CERTDIR* を定義した場合は、Adaptive Server によって *$SYBASE_CERTDIR/servername.txt* (UNIX の場合) または *%SYBASE_CERTDIR%¥servername.txt* (Windows の場合) から証明書がロードされます。
- *$SYBASE_CERTDIR* を定義しなかった場合は、Adaptive Server によって *$SYBASE/$SYBASE_ASE/certificates/servername.txt* (UNIX の場合) または *%SYBASE%¥%SYBASE_ASE%¥certificates¥servername.txt* (Windows の場合) から証明書がロードされます。

2 Adaptive Server を再起動することによって、信頼されたルート証明書ファイルを変更します。

3 sp_ldapadmin を使用し、*ldap://* 形式の URL ではなく *ldaps://* 形式の URL を指定して、LDAP サーバのセキュア・ポートへのセキュア接続を確立します。

4 次のいずれかの構文を使用して、プレイン・テキストでの TCP 接続を介して TLS セッションを確立します。

```
sp_ldapadmin 'starttls_on_primary', {true | false}
```

または

```
sp_ldapadmin 'starttls_on_secondary', {true | false}
```

注意 LDAP サーバ接続には connect timeout オプションがありません。LDAP サーバが応答を停止した場合は、すべてのログイン接続も応答を停止します。

LDAPS ユーザ認証の強化

以前のバージョンの Adaptive Server では、CA (認証局) によって信頼されたルート・ファイルに変更を加えた場合に、Adaptive Server を再起動して変更を有効にする必要があります。Adaptive Server バージョン 15.0.3 以降では、信頼されたルート・ファイルへの変更がサポートされているため、サーバを再起動する必要がありません。新しく追加されたサブコマンド reinit_descriptors は、LDAP サーバ記述子のバインドを解除して、ユーザ認証サブシステムを再初期化します。このオプションの構文については、『リファレンス・マニュアル：プロシージャ』を参照してください。

- このコマンドを実行するには、システム・セキュリティ担当者のパーミッションが必要です。
- システム・セキュリティ担当者のパーミッションを持つユーザが、このコマンドを実行しないで、信頼されたルート・ファイルを変更した場合、ハウスキーピング・ユーティリティのチョア・タスクでは、ユーザ認証サブシステムを 60 分ごとに再初期化するように設計された、新しいチョアが使用されます。

自動的な LDAP ユーザ認証とフェールバック

Adaptive Server 15.0.3 では、セカンダリ LDAP サーバがサポートされています。以前のバージョンでは、障害の発生したプライマリ LDAP サーバをオンライン状態にしたら、新しい LDAP ログインを認証して、プライマリ LDAP サーバに移動するために、LDAP サーバを手動でアクティブにする必要がありました。

バージョン 15.0.3 以降では、LDAP サーバを自動的にアクティブにするための、新しいチョア 'set_failback_interval' が Adaptive Server のハウスキーピング・ユーティリティに追加されています。構文については、「LDAP フェールバック時間間隔の設定」(524 ページ) を参照してください。

sp_ldapadmin set_failback_interval の set_failback_interval オプションは、障害の発生した LDAP サーバをアクティブにするための試行間隔を設定します。このパラメータを設定しない場合は、デフォルト値である 15 分が使用されます。『リファレンス・マニュアル：プロシージャ』の「sp_ldapadmin」を参照してください。

プライマリ URL のステータスが FAILED の場合、ハウスキーピング・タスクは、プライマリ・アクセス・アカウントの DN (識別名) とパスワードを使用して、プライマリ URL をアクティブにしようとします。プライマリ・アクセス・アカウントを設定していない場合、ハウスキーピング・タスクは匿名バインドの使用を試みます。初回の試行時にバインド操作が失敗した場合、ハウスキーピング・タスクは、設定された再試行回数だけバインド操作を再試行します。バインド操作が成功すると、プライマリ URL のステータスが READY になります。

セカンダリ URL のステータスが FAILED の場合、ハウスキーピング・タスクは、同様の方法でセカンダリ URL をアクティブにしようとします。

sp_ldapadmin の reinit_descriptors オプションは、証明書ファイルが変更されたときに実行されます。この場合、LDAP ユーザ認証サブシステムは 60 分ごとに再初期化されます。

フェールバック間隔がユーザによって設定されると、ハウスキーピング・タスクは、チョアを一掃するたびに、障害の発生した LDAP サーバの有無を調べます。障害の発生した LDAP サーバが見つかった場合は、フェールバック時間間隔で指定した時間が経過すると、LDAP サーバのアクティブ化が試みられます。

LDAP フェールバック時間間隔の設定

sp_ldapadmin set_failback_interval の構文は次のとおりです。

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

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)
```

例

この例では、LDAP フェールバック時間間隔が 60 分に設定されます。

```
sp_ldapadmin 'set_failback_interval' 60
```

この例では、LDAP フェールバック

時間間隔がデフォルト値である 15 分に設定されます。

```
sp_ldapadmin 'set_failback_interval' -1
```

この例では、フェールバック間隔の設定値が表示されます。

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

外部認証のログイン・マッピング

外部認証メカニズムを設定したときに、内部 Adaptive Server ログインに対する外部ユーザのマッピングが 1 つだけあり、認証が成功した場合、Adaptive Server は外部ユーザのパスワードと一致するように内部ログインのパスワードを更新します。たとえば、次のような状況が考えられます。

1　USER1 の Adaptive Server ログイン名は user_ase、パスワードは user_password です。もう 1 人のユーザの LDAP ログイン名は user_ldap、パスワードは

user_ldappasswd です。

2　Adaptive Server では、user_ldap と user_ase が一対一でマッピングされています。

3　ユーザ `user_ldap` は、パスワード `user_ldappasswd` を使用して Adaptive Server にログインします。

4　Adaptive Server は、パスワード `user_ase` を `user_ldapppasswd` に更新します。

次の例では、外部ユーザを Adaptive Server 内部ログインにマップすることによって外部認証メカニズムを設定した場合に、認証が Adaptive Server にフェールオーバすると、この外部ユーザの名前と正しい Adaptive Server パスワードを使用してログインできます。Adaptive Server は、マップされた内部ログインを内部で使用して、外部ユーザを認証します。

1 あるユーザの Adaptive Server ログイン名は user_ase、パスワードは user_password です。

2 もう 1 人のユーザの LDAP ログイン名は user_ldap です。

Adaptive Server は、user_ldap を user_ase にマップします。

3 次のように入力して LDAP を有効にします。

```
sp_configure 'enable ldap user auth', 1
```

4 この場合に、LDAP サーバがシャットダウンされるかクラッシュすると、user_ldap は、ログイン名 user_ldap とパスワード user_password を使用して Adaptive Server にログインできます。

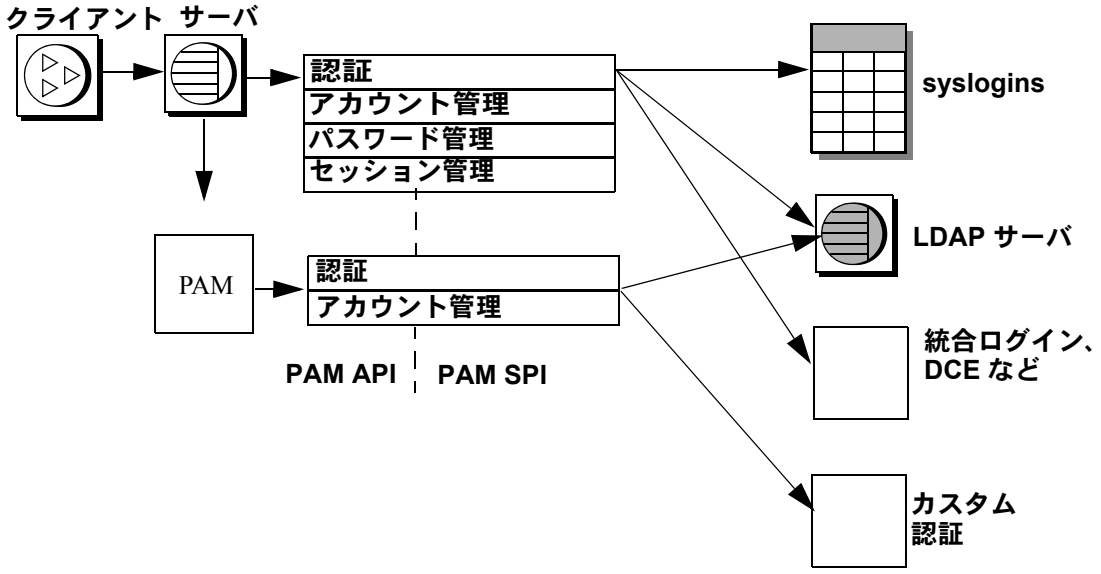
PAM を使用する認証のための Adaptive Server の設定

PAM (Pluggable Authentication Module) のサポートにより、認証を必要とするアプリケーションを変更せずに、複数の認証サービス・モジュールをまとめて使用できます。

PAM により Adaptive Server が Solaris や Linux のオペレーティング・システムに統合され、ユーザ・アカウントや認証メカニズムの管理が単純化され、総保有コスト (TCO) が削減されます。ユーザは、独自の認証モジュールや許可モジュールをカスタマイズしたり、作成したりできます。

注意 現在 PAM がサポートされているプラットフォームは Linux と Solaris です。PAM ユーザ認証の詳細については、各オペレーティング・システムのマニュアルを参照してください。

図 *16-2: PAM アーキテクチャ*



Adaptive Server は、ログイン・パケットから取得したログイン名とクレデンシャルを PAM API に渡します。PAM は、オペレーティング・システムの設定ファイルの指定に従ってサービス・プロバイダ・モジュールをロードし、認証プロセスを完了するための関数を呼び出します。

Adaptive Server での PAM の有効化

Linux と Solaris には定義済みの PAM モジュールがあります。これらのモジュールのいずれか一方を使用することも、独自のモジュールを作成することもできます。独自のモジュールを作成する場合は、オペレーティング・システムのマニュアルに記載されている PAM モジュールの作成に関する指示に従ってください。

注意　PAM モジュールを作成する場合は、RFC 86.0「Unified Login With Pluggable Authentication Modules (PAM」に準拠する必要があります。Adaptive Server では、RFC の認証管理モジュールがサポートされています。アカウント管理、セッション管理、またはパスワード管理のモジュールはサポートされていません。

オペレーティング・システムの設定

PAM サポートを有効にするには、各オペレーティング・システムを次のように設定します。

- Solaris では、*/etc/pam.conf* に次の行を追加します。

```
ase auth required /user/lib/security/$ISA/pam_unix.so.1
```

- Linux では、*/etc/pam.d/ase* という新しいファイルを作成して次の行を入力します。

```
auth requried /lib/security/pam_unix.so
```

これらのエントリの作成方法の詳細については、オペレーティング・システムのマニュアルを参照してください。

同一マシンでの 32 ビット・サーバと 64 ビット・サーバの実行

$ISA は、32 ビット・ライブラリと 64 ビット・ライブラリを同時に実行するために使用される環境変数です。

Solaris の 32 ビット・マシンでは $ISA は空文字列に置き換えられ、64 ビット・マシンでは文字列 “sparcv9” に置き換えられます。

32 ビット・サーバと 64 ビット・サーバの両方を使用する場合は、32 ビット版 PAM モジュールを任意のディレクトリに格納し、64 ビット版 PAM モジュールをそのディレクトリのサブディレクトリに格納します。

pam.conf のエントリは次のようになります。

```
$ ls /usr/lib/security/pam_sec.so.1
pam_sec.so.1 -> /SYBASE/pam_whatever_32bits.so.1

$ ls /usr/lib/security/sparcv9/pam_sec.so.1
pam_sec.so.1 -> /SYBASE/pam_sec_64bits.so.1

ase    auth    required /usr/lib/security/$ISA/pam_sec.so.1
```

注意 *pam.conf* に指定できる変数は $ISA のみです。

PAM ユーザ認証のための Adaptive Server の設定

enable pam user auth は、PAM ユーザ認証サポートを有効にします。

```
sp_configure "enable pam user auth", 0 | 1 | 2
```

各パラメータの意味は、次のとおりです。

- 0 － PAM 認証を無効にします。これがデフォルト値です。
- 1 － Adaptive Server は最初に PAM 認証を試行し、失敗した場合は syslogins 認証を使用します。
- 2 － PAM 認証のみを使用できるように指定します。

注意 PAM が有効な場合、パスワード管理は PAM サービス・プロバイダに委任されます。

Adaptive Server ログインと PAM ユーザ・アカウント

enable PAM user authentication を設定し、Adaptive Server とオペレーティング・システムの両方で PAM を設定したら、ユーザ・アカウントを設定します。オペレーティング・システム管理者またはネットワーク・セキュリティ管理者が PAM サービス・プロバイダのユーザ・アカウントの作成と管理を行い、データベース管理者が Adaptive Server のアカウントの作成と管理を行います。また、データベース管理者は、管理オプションを使用して、Adaptive Server と PAM サーバなどの外部認証メカニズムを統合するときのログイン・アカウントを柔軟に設定できます。データベース管理者は、従来のコマンドとプロシージャを使用して、Adaptive Server アカウントの役割、デフォルト・データベース、デフォルト言語、およびその他のログイン固有の属性の管理を続行できます。

表 16-15 は、ログイン時の syslogins テーブルの変更を示します。ここに示す変更は、PAM ユーザ認証が設定済みで、ログインが PAM の使用を制限されておらず、create login マッピングを設定していないことを前提としています。

表 *16-15: PAM* による *syslogins* の変更

syslogins にそのユーザのローが既に存在する	PAM 認証に成功	syslogins の変更
いいえ	はい	変更なし、ログインは失敗
いいえ	いいえ	変更なし、ログインは失敗
はい	はい	パスワードが変更された場合は、ローが更新される
はい	いいえ	変更なし

機能拡張されたログイン制御

前述の LDAP と PAM の項目で説明した方法に従って、サーバ全体の認証メカニズムを使用するように Adaptive Server を設定します。また、以下に説明する、機能拡張された Adaptive Server ログイン制御を使用して、サーバ上のログインごとに特定の認証メカニズムを使用するように指定することもできます。

ログインごとの制御は、サーバの認証メカニズムを移行中である場合や、ローカルなサーバ管理が必要で、集中管理されたユーザ・ログインに関連付けられていないサーバ固有のログインを制御する場合に便利です。

認証の強制

sp_modifylogin と sp_addlogin に対して以下のパラメータを使用して、ログインで特定の認証プロセスを使用するように強制できます。

- ASE − syslogins テーブルに格納されているパスワードを使用する Adaptive Server 内部認証を使用する。
- LDAP − LDAP サーバによる外部認証を使用する。
- PAM − PAM による外部認証を使用する。
- ANY − デフォルトのユーザ認証メソッド。ユーザに対して ANY 認証を指定すると、Adaptive Server は外部認証メカニズムが定義されているかどうかを調べます。定義されている場合は、そのメカニズムが使用されます。定義されていない場合は、Adaptive Server の認証が使用されます。

Adaptive Server は次の順序で外部認証メカニズムを調べます。

1 LDAP

2 PAM (Pluggable Authentication Module)。LDAP と PAM の両方が有効な場合、ユーザに対して PAM 認証は試行されない。

3 PAM と LDAP がどちらも有効になっていない場合は、syslogins によってログインが認証される。

“sa” などのログイン・アカウントは、引き続き syslogins カタログを使用して検証されます。ログインの認証を設定できるのは、SSO の役割を付与されているユーザだけです。

sp_modifylogin を使用してログインを認証する例を次に示します。

```
sp_modifylogin "nightlyjob", "authenticate with", "ASE"
sp_displaylogin "nightlyjob"
```

これによって次のような出力が表示されます。

```
Suid: 1234
Loginname:nightlyjob
Fullname:Batch Login
Default Database:master
. . .
Date of Last Password Change:Oct 2 2003 7:38 PM
Password expiration interval: 0
Password expired:N
Minimum password length:
Maximum failed logins: 0
Current failed login attempts:
Authenticate with:ASE
```

sp_maplogin を使用したログインのマッピング

次の構文で sp_maplogin を使用してログインをマップできます。

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

各パラメータの意味は、次のとおりです。

- authentication_mech － sp_modifylogin の authenticate with オプションに指定できる有効な値の 1 つ。
- *client_username* － 外部ユーザ名。オペレーティング・システム名、LDAP サーバのユーザ名、または PAM ライブラリが認識できる任意の名前を指定できます。null 値を指定すると、すべてのログイン名が有効になります。
- *action* － create login または drop を指定します。create login を使用すると、ログインが認証されると同時にログインが作成されます。drop はログインを削除するときに使用します。
- *login_name* は、syslogins に既に存在する Adaptive Server ログインです。

次の例は、外部ユーザ “jsmith” を Adaptive Server 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 Adaptive Server ユーザ名の両方が表示されます。

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

次の例は、LDAP で認証されたすべての外部ユーザについて、ログインが存在しない場合は新規ログインを作成するように Adaptive Server に指示します。

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

マッピング情報の表示

sp_helpmaplogin はマッピング情報を表示します。

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

authentication_mech は sp_modifylogin の authenticate with オプションに指定できる有効な値の 1 つで、*client_username* は外部ユーザ名です。

パラメータを指定せずに sp_helpmaplogin を使用した場合、Adaptive Server に現在ログインしているすべてのユーザに関するログイン情報が表示されます。上記のパラメータを使用すると、出力をクライアント・ユーザ名または認証メカニズムの特定のセットに限定できます。

次に、すべてのログインに関する情報を表示する例を示します。

```
sp_helpmaplogin
authentication   client name   login name
--------------   -----------   ------------------
NULL             jsmith        guest
LDAP             NULL           create login
```

認証メカニズムの設定

Adaptive Server で使用する認証メカニズムを設定するには、*@@authmech* グローバル変数を使用します。

たとえば、Adaptive Server でフェールオーバ対応の LDAP ユーザ認証が有効になっており (enable ldap user auth = 2) 、ユーザ “Joe” が ANY 認証の外部ユーザである場合、Joe がログインすると、Adaptive Server は LDAP ユーザ認証で Joe を認証しようとします。Joe の LDAP でのユーザ認証が失敗すると、Adaptive Server は Adaptive Server 認証を使用して Joe を認証します。これが成功するとログインできます。

この場合の *@@authmech* グローバル変数の値は次のとおりです。

```
select @@authmech
---------------------------------
ase
```

Adaptive Server が厳密な LDAP ユーザ認証を使用するように設定されており (enable ldap user auth = 2)、Joe が有効なユーザとして LDAP に追加された場合、Joe がログインするときの @@authmech の値は次のようになります。

```
select @@authmech---------------------------------
ldap
```

第 17 章

ユーザ・パーミッションの管理

この章では、ユーザ・パーミッションの使用と実装について説明します。

概要

「任意アクセス制御 **(DAC)**」を使用すると、ユーザの ID、グループのメンバシップ、アクティブな役割に基づいて、オブジェクトやコマンドに対するアクセスを制限できます。オブジェクト所有者などの特定のアクセス・パーミッションを持つユーザは、そのアクセス・パーミッションを他のユーザに渡すかどうかを選択できるので、制御は「任意」であると言えます。

Adaptive Server の任意アクセス制御システムは、次のタイプのユーザを識別します。

- システムで定義された 1 つまたは複数の役割を持つユーザ : 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、他の役割
- 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できます。

データベース所有者は、他のユーザが所有するオブジェクトに対するパーミッションを自動的に受け取るわけではありませんが、以下のことが実行できます。

- setuser コマンドを使用して、データベース内の特定のユーザの ID を持ち、そのユーザのすべてのパーミッションを一時的に取得する。
- setuser コマンドを使用してオブジェクト所有者の ID を持ち、次に grant コマンドを使用してパーミッションを付与することによって、そのオブジェクトに対するパーミッションを永続的に取得する。

別のユーザの ID を使用して、データベースまたはオブジェクトに対するそのユーザのパーミッションを取得する方法については、「別のユーザのパーミッションの取得」(560 ページ) を参照してください。

オブジェクト所有者は、オブジェクトへのアクセス権を他のユーザに付与したり、アクセス・パーミッションを他のユーザに渡す権限を他のユーザに付与したりすることもできます。grant コマンドを使用すると、ユーザ、グループ、役割に対して各種のパーミッションを与えることができます。また、revoke コマンドを使用するとパーミッションを無効にできます。grant と revoke コマンドを使用して、次のパーミッションをユーザに与えます。

- データベースの作成
- データベース内のオブジェクトの作成
- dbcc や set proxy などの特定のコマンドの実行
- 指定したテーブル、ビュー、ストアド・プロシージャ、暗号化キー、カラムへのアクセス

grant と revoke を使用して、システム・テーブルに対するパーミッションも設定できます。

デフォルトで “public” に与えられるパーミッションについては、grant 文や revoke 文の実行は不要です。

すべてのユーザがいつでもパーミッションなしで使用できるコマンドもあります。また、特定ステータスのユーザしか使用できず、譲渡できないコマンドもあります。

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

ビューとストアド・プロシージャをセキュリティ・メカニズムとして使用することもできます。「セキュリティ・メカニズムとしてのビューとストアド・プロシージャの使用」(570 ページ) を参照してください。

データベース作成用のパーミッション

create database コマンドを使用するパーミッションを付与できるのは、システム管理者だけです。create database のパーミッションを受け取るユーザは、master データベースの有効なユーザでもある必要があります。これは、データベースの作成は master を使用している状態で行われるためです。

多くのインストール環境では、システム管理者だけが create database パーミッションを持ち、データベースの配置とデータベース・デバイスの領域の割り付けを集中管理します。このような状況では、システム管理者が他のユーザに代わって新しいデータベースを作成し、所有権を該当するユーザに譲渡します。

別のユーザに所有させるデータベースを作成するには、次の手順に従います。

1　master データベース内で create database コマンドを発行します。

2　use コマンドを使用して、作成した新しいデータベースに切り替えます。

3　sp_changedbowner を実行します。

データベース所有権の変更

sp_changedbowner を使うと、データベースの所有権を変更できます。システム管理者はユーザ・データベースを作成して、必要な初期設定作業を完了してからその所有権を別のユーザに付与できます。sp_changedbowner を実行できるのはシステム管理者だけです。

ユーザをデータベースに追加する前、およびそのユーザによってデータベース内にオブジェクトが作成される前に所有権を譲渡することをおすすめします。新しい所有者は Adaptive Server 上に既にログイン名を持っている必要がありますが、そのデータベースのユーザであったりデータベースにエイリアスを持っていたりしてはなりません。そのような場合は、sp_dropuser または sp_dropalias を実行してからでなければ、データベースの所有権は変更できません。また、ユーザを削除する前に、オブジェクトの削除が必要なこともあります。

sp_changedbowner は、所有権を変更するデータベース内で発行します。構文は次のとおりです。

```
sp_changedbowner loginame [, true ]
```

ユーザ “albert” を現在のデータベースの所有者にして、元の “dbo” ユーザのエイリアスを削除する例を次に示します。

```
sp_changedbowner albert
```

エイリアスとそのパーミッションを新しい “dbo” に移動するには、true パラメータを指定します。

注意　master、model、tempdb、または sybsystemprocs のデータベースの所有権は変更できません。その他のシステム・データベースの所有権も変更しないでください。

データベース所有者の権限

オブジェクト作成パーミッションを他のユーザに付与できるのは、データベース所有者とシステム管理者だけです (ただし、暗号化キー作成およびトリガ作成パーミッションを付与できるのは、システム・セキュリティ担当者だけです)。データベース所有者は、そのデータベース内であらゆる作業を実行する権限を持っています。また、grant コマンドを使って他のユーザに明示的にパーミッションを付与しなければなりません。

次のコマンドを使用するためのパーミッションは、自動的にデータベース所有者に付与され、他のユーザに渡すことはできません。

- checkpoint
- dbcc
- alter database
- online database
- drop database
- dump database
- dump transaction
- grant (オブジェクト作成パーミッション)
- load database
- load transaction
- revoke (オブジェクト作成パーミッション)
- setuser

データベース所有者は、次のようにパーミッションの付与と取り消しを行うことができます。

- create default、create procedure、create rule、create table、create view の各コマンドの使用。

  データベース所有者は、sa_role を持ち、master データベースを使用している場合、create database、set tracing、および connect を使用するためのパーミッションを付与できます。

  データベース所有者は、sso_role を持っている場合、set session authorization、create trigger、および create encryption key 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。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。

- default permissions on system tables
- dbcc コマンドの使用：checkalloc、checkcatalog、checkdb、checkindex、checkstorage、checktable、checkverify、fix_text、indexalloc、reindex、tablealloc、textalloc、tune。

データベース・オブジェクト所有者

データベース・オブジェクト (テーブル、ビュー、暗号化キー、またはストアド・プロシージャ) を作成するユーザはそのオブジェクトの所有者となり、そのオブジェクトに対するすべてのオブジェクト・アクセス・パーミッションを自動的に付与されます。オブジェクト所有者以外のユーザ (データベースの所有者も含む) は、オブジェクト所有者またはそのオブジェクトに対するパーミッションを付与する grant パーミッションを持つユーザによって明示的にパーミッションを付与されないかぎり、そのオブジェクトに対するすべてのパーミッションを自動的に拒否されます。

たとえば、Mary が pubs2 データベースの所有者であり、そのデータベース内にテーブルを作成するためのパーミッションを Joe に付与したとします。Joe は、テーブル new_authors を作成し、このデータベース・オブジェクトの所有者になります。

初めは、new_authors のオブジェクト・アクセス・パーミッションを持つのは Joe だけです。Joe は、このテーブルに対するオブジェクト・アクセス・パーミッションを他のユーザに付与したり取り消したりできます。

次のオブジェクト変更パーミッションは、デフォルトではテーブルの所有者にあり、他のユーザに譲渡することはできません。

- alter table
- drop table
- create index

特定のデータベース・オブジェクトに対する select、insert、update、delete、references、decrypt、truncate table、update statistics、delete statistics、execute の各パーミッションを特定のユーザに付与する grant コマンドと revoke コマンドを使用するためのパーミッションは、grant with grant option コマンドを使って譲渡することができます。

オブジェクト (テーブル、ビュー、インデックス、ストアド・プロシージャ、ルール、暗号化キー、トリガ、またはデフォルト) を削除するための drop パーミッションは、デフォルトではオブジェクト所有者にあり、他のユーザには譲渡できません。

その他のデータベース・ユーザの権限

その他のデータベース・ユーザは、階層の最下部に位置付けられます。このデータベース・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、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。

システム・プロシージャに対するパーミッション

システム・プロシージャに対するパーミッションは、システム・プロシージャが格納されている sybsystemprocs データベースで設定します。

セキュリティ関連のシステム・プロシージャを実行できるのは、システム・セキュリティ担当者だけです。その他のシステム・プロシージャの中には、システム管理者しか実行できないものもあります。

また、データベース所有者しか実行できないシステム・プロシージャもあります。これらのプロシージャは、プロシージャを実行するユーザが、プロシージャの実行元であるデータベースの所有者であることを確認します。

その他のシステム・プロシージャは、パーミッションを付与されているユーザであれば実行できます。つまり、ユーザはシステム・プロシージャを実行するためのパーミッションをすべてのデータベースで持つか、あるいは、どのデータベースでも持たないかのどちらかです。

sybsystemprocs..sysusers に登録されていないユーザは、sybsystemprocs では “guest” として扱われ、多くのシステム・プロシージャに対するパーミッションを自動的に付与されます。システム・プロシージャに対するユーザのパーミッションを取り消すには、システム管理者がそのユーザを sybsystemprocs..sysusers に追加して、そのプロシージャに適用される revoke 文を発行する必要があります。ユーザ・データベースの所有者が自分のデータベースからシステム・プロシージャに対するパーミッションを直接制御することはできません。

パーミッションの付与と取り消し

grant と revoke を使用して、次のタイプのパーミッションを制御できます。

- オブジェクト・アクセス・パーミッション
- 関数から選択するパーミッション
- コマンドを実行するパーミッション (P)
- 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調整します。たとえば、authors テーブルに対して select コマンドを使用するには、パーミッションがユーザに明示的に付与されていなければなりません。オブジェクト・アクセス・パーミッションの付与と取り消しは、オブジェクト所有者 (およびシステム管理者またはシステム・セキュリティ担当者) が行います。オブジェクト所有者は、他のユーザにこのパーミッションを付与できます。

表 17-1 は、オブジェクト・アクセス・パーミッションのタイプと、そのパーミッションが適用されるオブジェクトを示します。

表 *17-1:* パーミッションと適用するオブジェクト

パーミッション	オブジェクト
select	テーブル、ビュー、カラム
update	テーブル、ビュー、カラム
insert	テーブル、ビュー
delete	テーブル、ビュー
references	テーブル、カラム
execute	ストアド・プロシージャ
truncate table	テーブル
delete statistics	テーブル
update statistics	テーブル
decrypt	テーブル、ビュー、カラム
select	暗号化キー

references パーミッションとは、alter table コマンドや create table コマンドで指定できる参照整合性制約のことです。decrypt パーミッションとは、暗号化カラムを復号化するために必要なパーミッションのことです。暗号化キーの select パーミッションとは、create table、alter table、または select into の各コマンドの暗号化キーを使用してカラムを暗号化するために必要なパーミッションのことです。それ以外のパーミッションは、SQL コマンドのことを指します。オブジェクト・アクセス・パーミッションは、デフォルトでは、オブジェクト所有者、システム管理者、または暗号化カラムの decrypt および暗号化キーの select に関するシステム・セキュリティ担当者にあり、他のユーザに付与できます。

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。構文は次のとおりです。

```
grant {all [privileges]| permission_list}
    on { table_name [(column_list)]
        | view_name [(column_list)]
        | stored_procedure_name}
    to {public | name_list | role_name}
    [with grant option]
```

オブジェクト・アクセス・パーミッションを取り消すには、revoke コマンドを使用します。構文は次のとおりです。

```
grant option for
    {all [privileges] | permission_list}
    on { table_name [(column_list)]
        | view_name [(column_list)]
        | stored_procedure_name}
    from {public | name_list | role_name}
    [cascade]
```

- all または all privileges は、指定のオブジェクトに適用できるすべてのパーミッションを付与することを示します。ただし、decrypt パーミッションを除きます。すべてのオブジェクト所有者は、all にオブジェクト名を指定して実行することによって、自分のオブジェクトに対するパーミッションを付与したり取り消したりできます。ストアド・プロシージャに対するパーミッションを付与または取り消す場合は、all は execute と同じです。

  注意 insert、update statistics、delete statistics、truncate table、delete の各パーミッションはカラムには適用されないので、カラム・リストを指定する場合にこれらのパーミッションをパーミッション・リストに含める (またはキーワード all を使用する) ことはできません。

- *permission_list* には、付与するパーミッションのリストを指定します。複数のパーミッションを指定する場合は、カンマで区切ってください。表 17-2 は各タイプのオブジェクトに付与できるアクセス・パーミッションを示します。

表 *17-2:* オブジェクト・アクセス・パーミッション

オブジェクト	permission_list に指定できるパーミッション
テーブルまたはビュー	select、insert、delete、update、references、truncate table、update statistics、decrypt、delete statistics references はテーブルに適用されるが、ビューには適用されない。その他のパーミッションはテーブルとビューのどちらにも適用される。update statistics、delete statistics、および truncate table はテーブルには適用されるがビューには適用されない。
カラム	select、update、references
ストアド・プロシージャ	execute
暗号化キー	select

カラムは、*permission_list* または *column_list* のどちらかに指定できますが、両方には指定できません。

- on は、パーミッションを付与または取り消すオブジェクトを指定します。パーミッションを付与または取り消しできるのは、一度に 1 つのテーブル、ビュー、暗号化キー、またはストアド・プロシージャ・オブジェクトだけです。一度に複数のカラムに対してパーミッションを付与または取り消すことはできますが、その場合はすべてのカラムが同じテーブルまたはビューに存在する必要があります。パーミッションの付与または取り消しができるのは、現在のデータベース内のオブジェクトに対してだけです。
- public は、Adaptive Server のすべてのユーザが含まれるグループ “public” のことです。public の意味は、grant と revoke とでは多少異なります。
  - grant の場合は、public にオブジェクト所有者が含まれます。したがって、自分のオブジェクトに対する自分のパーミッションを取り消し、その後で public に grant パーミッションを付与すると、“public” の他のユーザと同様に、自分にもそのパーミッションが再び与えられます。
  - revoke の場合は、public に所有者は含まれません。
- *name_list* には、次のものを指定します。
  - グループ名
  - ユーザ名
  - ユーザ名とグループ名の組み合わせ。それぞれの名前はカンマで区切ります。
- *role_name* は、Adaptive Server のシステム定義の役割またはユーザ定義の役割の名前です。ユーザ定義の役割の階層を作成して定義し、付与されている特定の役割に基づいて、権限を付与することができます。システム定義の役割には、sa_role (システム管理者)、sso_role (システム・セキュリティ担当者)、oper_role (オペレータ) があります。システム定義の役割は、作成も変更もできません。
- grant 文で with grant option を使用すると、*name_list* に指定されているユーザが、指定されたオブジェクト・アクセス・パーミッションを他のユーザに付与できるようになります。ユーザがオブジェクトに対して with grant option パーミッションを持っている場合、そのオブジェクトに対するパーミッションが public またはそのユーザが属するグループから取り消されても、そのユーザの with grant option パーミッションは取り消されません。

- grant option for を指定すると、with grant option パーミッションが取り消されます。つまり、*name_list* に指定されているユーザは、指定されたパーミッションを他のユーザに付与できなくなります。パーミッションを付与されていたユーザによって別のユーザにもパーミッションが付与されている場合は、cascade オプションを使用して、それらのユーザからもパーミッションを取り消す必要があります。*name_list* に指定されているユーザは、オブジェクトにアクセスするためのパーミッションは保持しますが、他のユーザにアクセス権を付与することはできなくなります。grant option for は、オブジェクト・アクセス・パーミッションだけに適用され、オブジェクト作成パーミッションには適用されません。
- revoke 文の cascade オプションは、指定されたオブジェクト・アクセス・パーミッションを *name_list* に指定されているユーザから削除し、さらにそのユーザからパーミッションの付与を受けたすべてのユーザからも削除します。

付与または取り消しができるのは、現在のデータベース内のオブジェクトに対するパーミッションだけです。

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アクセス権を取り消すか、またはシステム管理者が取り消すまで有効です。つまり、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、そのユーザによるアクセスは拒否されます。

暗号化キー作成パーミッションの付与や取り消しを実行できるのは、システム・セキュリティ担当者だけです。データベースの所有者は、どのユーザ・テーブルにもトリガを作成できます。ユーザは、各自が所有しているテーブルにのみトリガを作成できます。

create trigger コマンドを発行するパーミッションは、デフォルトでユーザに付与されます。

ユーザがトリガを作成するためのパーミッションが、システム・セキュリティ管理者によって取り消されると、sysprotects テーブルにそのユーザに対する取り消しローが追加されます。create trigger コマンドを発行するパーミッションをそのユーザに付与するには、grant コマンドを 2 回発行する必要があります。最初のコマンドで sysprotects テーブルから取り消しローを削除し、2 番目のコマンドで付与ローを挿入します。トリガを作成するには、システム・セキュリティ担当者がパーミッションを付与する必要があります。トリガを作成するパーミッションが取り消されると、ユーザは自分が所有するテーブルであってもトリガの作成はできなくなります。ユーザに対するトリガ作成パーミッションの取り消しの影響を受けるのは、revoke コマンドが発行されたデータベースだけです。

具体的 ID

Adaptive Server は、セッション中のユーザをログイン名によって識別します。この識別は、サーバのすべてのデータベースで有効です。ユーザがオブジェクトを作成すると、所有者のデータベース・ユーザ ID (*uid*) と作成者のログイン名の両方が、sysobjects テーブル内でオブジェクトと関連付けられます。この情報によって、どのユーザが所有するオブジェクトであるかが具体的に識別されるため、サーバは、いつオブジェクトのパーミッションが暗黙的に許可できるかを認識できます。

Adaptive Server ユーザがテーブルを作成し、そのテーブルにアクセスするプロシージャを作成したとき、そのプロシージャを使用するパーミッションを付与されたユーザには、そのオブジェクトに直接アクセスするためのパーミッションは必要ありません。たとえば、次のように “mary” というユーザに proc1 に対するパーミッションを付与したとき、mary はテーブル table1 に対する選択パーミッションは明示的に与えられてはいませんが、このテーブルのカラム id と descr を参照できます。

```
create table table1 (id     int,
                     amount money,
                     descr    varchar(100))

create procedure proc1 as select id, descr from table1

grant execute on proc1 to mary
```

ただし、オブジェクトを具体的に識別できる場合にのみ暗黙的パーミッションが有効となることもあります。たとえば、エイリアスとデータベース間オブジェクト・アクセスの両方が関係する場合です。

SQL92 標準に準拠するための要件

set コマンドを使用して ansi_permissions を on にした場合は、update 文と delete 文を実行するための追加のパーミッションが必要です。表 17-3 は、必要となるパーミッションをまとめたものです。

表 *17-3: update と delete に必要な ANSI パーミッション*

	必要なパーミッション：*set ansi_permissions off*	**必要なパーミッション：*set ansi_permissions on***
update	値を設定するカラムに対する update パーミッション	値を設定するカラムに対する update パーミッション および where 句に指定するすべてのカラムに対する select パーミッション set 句の右側のすべてのカラムに対する select パーミッション
delete	テーブルに対する delete パーミッション	ローを削除するテーブルに対する delete パーミッション および where 句に指定するすべてのカラムに対する select パーミッション

ansi_permissions が on の場合に、必要となる追加の select パーミッションが与えられていないユーザが更新または削除を行うと、トランザクションはロールバックされ、エラー・メッセージが表示されます。このエラー・メッセージが表示された場合は、すべての関係するカラムに対する select パーミッションをオブジェクト所有者が付与する必要があります。

オブジェクト・アクセス・パーミッションの付与の例

次の文は、titles テーブルに対して挿入と削除を行うためのパーミッションを Mary と “sales” グループに付与します。

```
grant insert, delete
on titles
to mary, sales
```

次の文はストアド・プロシージャ makelist を使用するためのパーミッションを Harold に付与します。

```
grant execute
on makelist
to harold
```

次の文は、カスタム・ストアド・プロシージャ sa_only_proc を実行するためのパーミッションを、システム管理者の役割を付与されているユーザに付与します。

```
grant execute
on sa_only_proc
to sa_role
```

次の文は、authors テーブルに対して選択、更新、削除を行うためのパーミッションと、他のユーザに同じパーミッションを付与するためのパーミッションを Aubrey に付与します。

```
grant select, update, delete
on authors
to aubrey
with grant option
```

オブジェクト・アクセス・パーミッションの取り消しの例

次の 2 つの文はどちらも、titles テーブルの price カラムと total_sales カラムを更新するためのパーミッションをテーブル所有者以外のすべてのユーザから取り消します。

```
revoke update
on titles (price, total_sales)
from public
```

次の文は、authors テーブルを更新するためのパーミッションを Clare から取り消すと同時に、Clare がそのパーミッションを付与したすべてのユーザからもそのパーミッションを取り消します。

```
revoke update
on authors
from clare
cascade
```

次の文は、カスタム・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
```

update statistics、*delete statistics*、*truncate table* のパーミッションの付与と取り消し

Adaptive Server では、update statistics、delete statistics、truncate table の各コマンドに対する、ユーザ、役割、グループのパーミッションを付与または取り消すことができます。テーブル所有者も、暗黙の grant によってパーミッションを付与できます。具体的には、update statistics、delete statistics、truncate table をストアド・プロシージャに追加してから、そのストアド・プロシージャの実行パーミッションをユーザまたは役割に付与します。

update statistics のパーミッションをカラム・レベルで付与または取り消すことはできません。sysroles、syssrvroles、sysloginroles の各セキュリティ・テーブルに対して update statistics または delete statistics を実行するには、sso_role が必要です。

デフォルトでは、sa_role を持つユーザは、sysroles、syssrvroles、sysloginroles 以外のシステム・テーブルに対して update statistics と delete statistics を実行するパーミッションがあり、この権限を他のユーザに渡すこともできます。

grant と revoke の構文の一部は次のとおりです。

grant [truncate table | update statistics | delete statistics] on *table_name* to {*user_name* | *role_name* | *group_name*}

revoke [truncate table | update statistics | delete statistics] on *table_name* from {*user_name* | *role_name* | *group_name*}

grant all を発行して、update statistics、delete statistics、truncate table のパーミッションを付与することもできます。

たとえば、次の例は、ユーザ “harry” が authors テーブルに対して truncate table と updates statistics を使用できるようにします。

```
grant truncate table on authors to harry
grant update statistics on authors to harry
```

次の例は、“harry” の authors テーブルに対する truncate table 権限と update statistics 権限を取り消します。

```
revoke truncate table on authors from harry
revoke update statistics on authors from harry
```

次の例は、ユーザ “billy” が authors テーブルに対して delete statistics コマンドを使用できるようにします。

```
grant delete statistics on authors to billy
```

次の例は、ユーザ “billy” の authors テーブルに対する delete statistics 権限を取り消します。

```
revoke delete statistics on authors from billy
```

次の例は、oper_role を持つすべてのユーザに truncate table、update、delete statistics の各権限を付与します (ユーザ “billy” と “harry” は、oper_role を持っている場合、これらのコマンドを authors に対して実行できるようになります)。

```
grant truncate table on authors to oper_role
grant update statistics on authors to oper_role
grant delete statistics on authors to oper_role
```

次の例は、oper_role を持つすべてのユーザの truncate table、update statistics、delete statistics の各権限を取り消します。

```
revoke truncate table on authors from oper_role
revoke update statistics on authors from oper_role
revoke delete statistics on authors from oper_role
```

ユーザ “billy” と “harry” は、これらのコマンドを authors に対して実行できなくなります。

また、ストアド・プロシージャを使用して、truncate table、delete statistics、update statistics のパーミッションを暗黙に付与することもできます。たとえば、“billy” が authors テーブルを所有している場合、billy は次を実行すると、authors に対して truncate table と update statistics を実行する権限を 塗 arry・に付与できます。

```
create procedure sproc1
as
truncate table authors
update statistics authors
go
grant execute on sproc1 to harry
go
```

また、ストアド・プロシージャを使用してカラム・レベルで update statistics と delete statistics のパーミッションを暗黙に付与することもできます。

注意 update statistics を実行するパーミッションをユーザに付与すると、付与されたユーザはコマンドのバリエーション (update all statistics、update partition statistics、update index statistics、update statistics table など) を実行するパーミッションも取得します。たとえば、次の例は、authors テーブルに対して update statistics のすべてのバリエーションを実行するパーミッションを “billy” に付与します。

```
grant update statistics on authors to billy
```

update statistics を実行するパーミッションをユーザから取り消すと、そのコマンドのバリエーションを実行するパーミッションも取り消すことになります。

---

update statistics のバリエーション (update index statistics など) のパーミッションを個別に付与することはできません。つまり、次のようなコマンドは発行できません。

```
grant update all statistics to harry
```

ただし、ストアド・プロシージャを作成して、これらのコマンドをどのユーザが実行するかを制御することができます。たとえば、次の例は、authors テーブルに対して update index statistics を実行するパーミッションを “billy” に付与します。

```
create proc sp_ups as
update index statistics on authors
go
revoke update statistics on authors from billy
go
grant execute on sp_ups to billy
```

delete statistics のパーミッションをカラム・レベルで付与または取り消すことはできません。

Adaptive Server は、その他のグローバルな監査として truncate table を監査しますが、update statistics の監査は行いません。truncate table と update statistics の両方について明確な監査証跡を保持するためには、上記のように実行パーミッションをユーザに付与するストアド・プロシージャに両方のコマンドを含めることをおすすめします。

次の条件が当てはまり、かつユーザが update statistics、delete statistics、または truncate table コマンドを発行した場合、コマンドが失敗してエラー番号 10330 が生成されます。

- ユーザがテーブルを所有していない。
- ユーザが sa_role を持っていない。
- ユーザが、テーブルの所有者であるユーザになる setuser を使用したデータベースの所有者ではない。
- ユーザが、update statistics、delete statistics、または truncate table 権限を付与されていない。

関数のパーミッションの付与

grant select on builtin *function_name* を使用して、関数 set_appcontext、get_appcontext、list_appcontext、rm_appcontext を使用するパーミッションをユーザに付与します。

構文は次のとおりです。

```
grant select on [builtin] function_name
to { name_list | role_list }
```

各カラムの内容は、次のとおりです。

- builtin – 同じ名前のテーブルと付与可能な関数とを区別します。
- *function_name* – パーミッションを付与する関数の名前です。select パーミッションを付与できる関数は、set_appcontext、get_appcontext、list_appcontext、rm_appcontext です。
- *name_list* – ユーザのデータベース名とグループ名のリストです。
- *role_list* – パーミッションを付与するシステム定義またはユーザ定義の役割の名前のリストです。変数は使用できません。

get_appcontext 関数の select パーミッションを public に付与する場合は、次のように入力します。

```
grant select on builtin get_appcontext to public
```

コマンドを実行するパーミッションの付与と取り消し

この項では、特定のコマンドを実行するユーザのパーミッションを付与する方法と取り消す方法について説明します。

コマンドを実行するパーミッションの付与

オブジェクト作成パーミッションは、オブジェクトを作成するコマンドの使用を調整します。connect、set session authorization など、オブジェクト作成以外のコマンドを付与できます。これらのパーミッションを付与できるのは、(特に明記されていないかぎり) システム管理者とデータベース所有者だけです。

コマンドは次のとおりです。

- connect
- create database
- create default
- create procedure
- create rule
- create table
- create view
- set session authorization
- create encryption key (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のみ)
- 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のみ)

コマンド・パーミッションの構文は、オブジェクト・アクセス・パーミッションの構文とわずかに違います。オブジェクト作成パーミッションの grant の構文を次に示します。

```
grant {all [privileges] | command_list}
    to {public | name_list | role_name}
```

オブジェクト作成パーミッションの revoke の構文を次に示します。

```
revoke {all [privileges] | command_list}
    from {public | name_list | role_name}
```

各パラメータの意味は、次のとおりです。

- all または all privileges を使用できるのは、システム管理者とデータベース所有者だけです。システム管理者が master データベース内で grant all を実行した場合は、create database も含めて、(create encryption key および create trigger 以外の) すべての create パーミッションが付与されます。システム管理者が master 以外のデータベースから grant all を実行した場合は、create database、create trigger、および create encryption key 以外のすべての create パーミッションが付与されます。データベース所有者が grant 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 パーミッションが付与され、Adaptive Server の情報メッセージが出力されます。

- *command_list* には、付与または取り消すオブジェクト作成および他のコマンドのパーミッションを指定します。複数のコマンドを指定する場合は、カンマで区切ってください。リストに指定できるのは、create database、create default、create procedure、create rule、create table、connect、create encryption key、set session authorization、create view、create trigger です。create database パーミッションは、システム管理者だけが付与できます。また、master データベース内からのみ付与できます。create encryption key、set session authorization、および create trigger パーミッションを付与するには、システム・セキュリティ担当者の権限が必要です。
- public は、データベース所有者以外のすべてのユーザです (データベース所有者は、そのデータベース内でのオブジェクト作成パーミッションを「所有」します)。
- *name_list* には、ユーザ名またはグループ名のリストをカンマで区切って指定します。
- *role_name* は、Adaptive Server システム標準の役割またはユーザ定義の役割の名前です。ユーザ定義の役割の階層を作成して定義し、付与されている特定の役割に基づいて、権限を付与することができます。

コマンドのパーミッションの付与の例

最初の例では、create database と create table を実行するためのパーミッションを Mary と John に付与します。create database パーミッションが付与されるため、このコマンドは master データベース内にいるシステム管理者だけが実行できます。Mary と John の create table パーミッションは、master データベースのみに適用されます。

```
grant create table, create database
to mary, john
```

次の文は、現在のデータベース内で create table と create view を実行するためのパーミッションをすべてのユーザに付与します。

```
grant create table, create view
to public
```

コマンドのパーミッションの取り消しの例

次の例では、テーブルとルールを作成するための “mary” のパーミッションを取り消します。

```
revoke create table, create rule
from mary
```

代理権限の付与

システム・セキュリティ担当者は、grant set proxy コマンドや grant set session authorization コマンドを使用して、サーバ内の別のユーザになり代わるためのパーミッションをユーザに与えます。このパーミッションを与えられたユーザは、set proxy または set session authorization を実行して、別のユーザになることができます。

代理権限使用の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、master データベースから grant コマンドを実行して付与します。構文は次のとおりです。

```
grant set proxy to user | role
    [restricted role role_list | all | system]
```

各パラメータの意味は、次のとおりです。

- *role_list* − ターゲット・ログインに対して制限する役割のリスト。付与されるユーザに *role_list* の役割がまだ付与されておらず、ターゲットのログインに *role_list* の役割があると、ターゲットのログインへの set proxy は失敗します。
- *all* − *role_list* に set proxy を付与すると、付与されたユーザが ID を切り替えたときに新しい役割が付与されなくなります。
- *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。

例 1

例 1：この例は、set proxy をユーザ “joe” に付与しますが、“joe” が ID を、sa_role、sso_role、または admin_role の役割を持つユーザに切り替えることは制限します (ただし、“joe” が既にこれらの役割を持っている場合は、これらの役割を持つユーザに対して set proxy を実行できます)。

```
grant set proxy to joe
restricted 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 を持っていないかぎりコマンドは失敗します。

```
set proxy Our_admin_role
Msg 10368, Level 14, State 1:
Server 's', Line 2: 自分にない役割がターゲット・ログインに含まれ、その使用を制限されているために、Set session 権限のパーミッションが拒否されました。
```

“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
```

例 2

例 2：ID を切り替えるときに “joe” に新しい役割が付与されないようにします。

```
grant set proxy to joe
restricted role all
```

“joe” は、自分と同じ役割 (または役割のサブセット) を持つユーザにしか set proxy を付与できません。

例 3

例 3 : set proxy を使用するときに Joe が新しいシステム役割を取得できないようにします。

```
grant set proxy to joe
restricted role system
```

joe が持っていないシステム役割をターゲット・ログインが持っていると、set proxy は失敗します。

dbcc コマンドのパーミッションの付与

システム管理者は、Adaptive Server のシステム管理者レベルの権限を持たないユーザや役割に対して、dbcc コマンドを実行するパーミッションを付与できます。この「任意アクセス制御」により、システム管理者はデータベース・オブジェクトまたは特定のデータベース・レベルとサーバ・レベルのアクションへのアクセスを制御できます。

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

サーバワイドとデータベース固有の *dbcc* コマンド

dbcc コマンドは、次のいずれかです。

- データベース固有 － 特定のターゲット・データベースに対して実行する dbcc コマンド (checkalloc、checktable、checkindex、checkstorage、checkdb、checkcatalog、checkverify、fix_text、indexalloc、reindex、tablealloc、textalloc など)。これらのコマンドは特定のデータベースを対象としたコマンドですが、パーミッションの付与や取り消しができるのはシステム管理者だけです。
- サーバワイド － tune コマンドなど、サーバ全体に作用するが、特定のデータベースには関連付けられていない dbcc コマンド。これらのコマンドのパーミッションはデフォルトでサーバワイドに付与され、どのデータベースにも関連付けられません。

システム管理者は、これらのデータベース内で有効なユーザとして設定することで、すべてのデータベースで dbcc コマンドを実行するパーミッションをそのユーザに付与できます。ただし、grant dbcc コマンドのパーミッションをユーザに個別に付与すると、各ユーザを手動でデータベースに追加しなければなりません。パーミッションを役割に対して付与すれば、ユーザは “guest” ユーザとしてデータベースを使用できるようになるので、こちらの方法がより便利です。

セキュリティ管理の観点から、データベース固有の dbcc コマンドのパーミッションをサーバワイドに付与する方法をシステム管理者が選ぶこともあります。たとえば、すべてのデータベースに対する grant dbcc checkstorage を storage_admin_role というユーザ定義の役割に対して実行すれば、storage_admin_role に対する grant dbcc checkstorage をデータベースごとに実行する手間が省けます。

次のコマンドは、サーバワイドで有効なコマンドですが、データベース固有のコマンドではありません。

- tune などのサーバワイド dbcc コマンド
- storage_admin_role に対して付与される grant dbcc checkstorage など、サーバワイドにパーミッションが付与されるデータベース固有の dbcc コマンド

dbcc コマンドのパーミッションの付与対象者とデータベース内のユーザ

grant dbcc コマンドと revoke dbcc コマンドは、データベース内のユーザに対して機能します。

データベース内の役割に対して初めて grant が実行されると、その役割は自動的にユーザとして追加されるため、役割に dbcc の権限を付与するための追加の要件はありません。ログインは、パーミッションが付与されるデータベース内の有効なユーザでなければなりません。有効なユーザには “guest” が含まれます。

サーバワイドな dbcc コマンドの場合、ログインは master データベース内の有効なユーザでなければなりません。また、システム管理者はパーミッションの付与を master データベース内から実行する必要があります。

データベース固有の dbcc コマンドの場合、ログインはターゲット・データベース内の有効なユーザでなければなりません。

システム・テーブルのパーミッション

システム・テーブルで使うパーミッションは、他のテーブルのパーミッションと同じくデータベース所有者が制御できます。データベースを作成すると、一部のシステム・テーブルの select パーミッションが public に付与され、一部のシステム・テーブルの select パーミッションが管理者に制限されます。テーブルによっては、いくつかのカラムで、public に対する select パーミッションが制限されている場合もあります。

特定のシステム・テーブルに対する現在のパーミッションを調べるには、次のように実行します。

```
sp_helprotect system_table_name
```

たとえば、master データベースの syssrvroles のパーミッションを調べるには、次のコマンドを実行します。

```
use master
go
sp_helprotect syssrvroles
go
```

デフォルトでは、データベース所有者も含め、ユーザがシステム・テーブルを直接変更することはできません。代わりに、T-SQL コマンドと Adaptive Server に付属するシステム・プロシージャを使用してシステム・テーブルを変更します。これは整合性の保証に役立ちます。

警告！ Adaptive Server にはシステム・テーブルを変更できるメカニズムがありますが、システム・テーブルの変更はしないことを強くおすすめします。

システム・テーブルとストアド・プロシージャへのデフォルト・パーミッションの付与

grant コマンドと revoke コマンドでは、default permissions パラメータを指定できます。installmodel または installmaster では、システム・テーブル (次の表を参照) のデフォルト・パーミッションは付与されません。代わりに、Adaptive Server が新しいデータベースを構築するときに、これらのシステム・テーブルのデフォルト・パーミッションが割り当てられます。構文の一部は次のとおりです。

```
grant default permissions on system tables
revoke default permissions on system tables
```

default permissions on system tables は、任意のデータベースからこのコマンドを発行するときに、次のシステム・テーブルのデフォルト・パーミッションの付与または取り消しを指定します。

sysalternates	sysjars	sysqueryplans	systypes
sysattributes	syskeys	sysreferences	sysusermessages
syscolumns	syslogs	sysroles	sysusers
syscomments	sysobjects	syssegments	sysxtypes
sysconstraints	syspartitions	sysstatistics	
sysdepends	sysprocedures	systabstats	
sysindexes	sysprotects	systhresholds	

default permissions on system tables では、次の変更も行われています。

- 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	sysremotelogins	systransactions
sysconfigures	syslocks	sysresourcelimits	sysusages
syscurconfigs	syslogins	sysservers	
sysdatabases	sysmessages	syssessions	
sysdevices	sysprocesses	systimeranges	

このコマンドでは次の変更も行われています。

- public から sysdatabases(audflags) の select を取り消す。
- public から syscolumns(encrkeyid) の select を取り消す。
- public から syscolumns(encrkeydb) の select を取り消す。
- sso_role に syscolumns の select を付与する。
- public から sysdatabases(deftabaud) の select を取り消す。
- public から sysdatabases(defvwaud) の select を取り消す。
- public から sysdatabases(defpraud) の select を取り消す。
- public から sysdatabases(audflags2) の select を取り消す。
- sysdatabases に対する select を sso_role に付与する。
- public から syslogins(password) の select を取り消す。
- public から syslogins(audflags) の select を取り消す。
- syslogins に対する select を sso_role に付与する。
- public から syslisteners(net_type) の select を取り消す。
- public から syslisteners(address_info) の select を取り消す。
- syslisteners に対する select を sso_role に付与する。
- public から syssrvroles(srid) の select を取り消す。
- public から syssrvroles(name) の select を取り消す。
- public から syssrvroles(password) の select を取り消す。
- public から syssrvroles(pwdate) の select を取り消す。
- public から syssrvroles(status) の select を取り消す。

- public から syssrvroles(logincount) の select を取り消す。
- syssrvroles に対する select を sso_role に付与する。
- public から sysloginroles(suid) の select を取り消す。
- public から sysloginroles(srid) の select を取り消す。
- public から sysloginroles(status) の select を取り消す。
- sso_role から sysloginroles に対する select を取り消す。

grant 文と *revoke* 文の組み合わせ

特定のパーミッションを特定のユーザに割り当てることができますが、ほとんどのユーザにほとんどの権限を付与するのであれば、すべてのユーザにすべてのパーミッションを付与してから、特定のユーザから特定のパーミッションを取り消す方が簡単です。

たとえば、データベース所有者は次の文を発行することによって、titles テーブルに対するすべてのパーミッションをすべてのユーザに付与できます。

```
grant all
on titles
to public
```

次に、次のような一連の revoke 文を発行します。

```
revoke update
on titles (price, advance)
from public
revoke delete
on titles
from mary, sales, john
```

grant 文と revoke 文の結果は、実行する順序によって異なります。競合が発生した場合は、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について

grant 文と revoke 文は発行順序が重要です。たとえば、titles テーブルに対する select パーミッションが Jose のグループに付与された後で、advance カラムを選択するための Jose のパーミッションが取り消された場合に、Jose が選択できるのは advance 以外のすべてのカラムですが、Jose と同じグループの他のユーザはこの場合もすべてのカラムを選択できます。

グループまたは役割に適用される grant 文や revoke 文は、そのグループまたは役割のメンバに割り当てられている競合するパーミッションを変更します。たとえば、titles テーブルの所有者が sales グループのメンバごとに異なるパーミッションを付与した後で、sales グループのメンバ全員に同じパーミッションを付与することにしたとします。その所有者は次の文を発行します。

```
revoke all on titles from sales
grant select on titles(title, title_id, type,
        pub_id)
    to sales
```

同じように、public に対して発行された grant 文と revoke 文は、以前に発行されたパーミッションの中で新しい状況と競合するすべてのパーミッションを、すべてのユーザについて変更します。

同じ grant 文と revoke 文でも、発行順序が異なると、結果もまったく異なります。たとえば、次の順序でこれらの文を発行すると、public グループに属する Jose は titles に対する select パーミッションを持たなくなります。

```
grant select on titles(title_id, title) to jose
revoke select on titles from public
```

これに対して、同じ文を逆の順序で発行すると、title_id と title カラムだけに対する select パーミッションを Jose だけが持つようになります。

```
revoke select on titles from public
grant select on titles(title_id, title) to jose
```

grant にキーワード public を使用した場合は、自分自身も含まれることを忘れないでください。オブジェクト作成パーミッションに対して revoke を実行するユーザは、データベース所有者でなければ public に含まれます。オブジェクト・アクセス・パーミッションに対して revoke を実行するユーザは、オブジェクト所有者でなければ public に含まれます。自分のテーブルを使用するための自分のパーミッションを取り消す一方で、そのテーブル上に作成されたビューにアクセスするためのパーミッションを自分自身に付与することもできます。このようにするには、grant 文と revoke 文を発行して明示的に自分のパーミッションを設定する必要があります。方針が変わった場合は、grant 文を使っていつでもパーミッションを再設定できます。

grant dbcc および set proxy の fipsflagger に対する警告の発行

set fipsflagger オプションが有効になっているときに grant dbcc と set proxy を発行すると、次の警告が発行されます。

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

役割の付与と取り消し

定義した役割は、サーバ内の任意のログイン・アカウントまたは役割に付与できます。ただし、相互排他性と階層の規則に違反しない場合にかぎります。表 17-4 は、役割に関連するタスクとそのタスクの実行に必要な役割、および使用するコマンドを示します。

表 *17-4:* タスク、必要な役割、および使用するコマンド

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
ユーザ定義の役割の付与	システム・セキュリティ担当者	grant role
役割階層の作成	システム・セキュリティ担当者	grant role
役割階層の変更	システム・セキュリティ担当者	revoke role
システム標準の役割の取り消し	システム・セキュリティ担当者	revoke role
ユーザ定義の役割の取り消し	システム・セキュリティ担当者	revoke role

役割の付与

ユーザまたは他の役割に役割を付与するには、次の構文を使用します。

```
grant role role_granted [{, role_granted}...]
    to grantee [{, grantee}...]
```

各パラメータの意味は、次のとおりです。

- *role_granted* は、付与する役割です。付与する役割は、いくつでも指定できます。
- *grantee* は、ユーザまたは役割の名前です。付与対象のユーザや役割はいくつでも指定できます。

grant 文で指定したすべての役割が、指定したすべてのユーザと役割に付与されます。ある役割を別の役割に付与すると、役割の階層が作成されます。

たとえば、Susan、Mary、John に役割 “financial_analyst” と “payroll_specialist” を付与するには、次のように入力します。

```
grant role financial_analyst, payroll_specialist
  to susan, mary, john
```

grant と役割について

grant コマンドを使用すると、システム標準の役割かユーザ定義の役割かに関係なく、指定した役割を付与されているすべてのユーザにオブジェクトのパーミッションを付与できます。これによって、次に示す役割を付与されているユーザに対してオブジェクトの使用を制限できます。

- システム標準の役割
- ユーザ定義の役割

役割は、ログイン・アカウントまたは別の役割に対してのみ付与できます。

しかし、grant パーミッションによる方法では、指定された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は防止できません。たとえば、あるストアド・プロシージャをシステム管理者だけが正しく実行できるようにする場合は、そのストアド・プロシージャ内で proc_role システム関数を使用します。詳細については、「役割に関する情報の表示」(416 ページ) を参照してください。

役割に付与されているパーミッションは、ユーザやグループに付与されているパーミッションよりも優先されます。たとえば、John がシステム・セキュリティ担当者の役割を付与されていて、sales テーブルに対するパーミッションが sso_role に付与されているとします。sales に対する John 個人のパーミッションが取り消されても、役割に付与されているパーミッションが個人に付与されているパーミッションよりも優先されるので、John は、sso_role をアクティブにすれば sales にアクセスできます。

パーミッションを付与するとき、システム管理者はオブジェクト所有者として扱われます。システム管理者が、別のユーザのオブジェクトに対するパーミッションを付与すると、sysprotects と sp_helprotect の出力では、オブジェクト所有者の名前が付与者として表示されます。

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。

役割の取り消し

revoke role を使用すると、ユーザや他の役割から役割を取り消すことができます。

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

各パラメータの意味は、次のとおりです。

- *role_name* は、取り消す役割の名前です。取り消す役割は、いくつでも指定できます。
- *grantee* は、ユーザまたは役割の名前です。付与対象のユーザや役割はいくつでも指定できます。

revoke 文で指定したすべての役割が、指定したすべてのユーザと役割から取り消されます。

ユーザがログインしている間は、そのユーザから役割を取り消すことはできません。

別のユーザのパーミッションの取得

Adaptive Server には、別のユーザの ID とパーミッション・ステータスを取得する方法が 2 つあります。

- データベース所有者は、setuser コマンドを使用して、現在のデータベース内の別のユーザになり代わり、その ID とパーミッション・ステータスを利用することができます。「setuser の使用」(560 ページ) を参照してください。
- 「代理権限」を利用すると、1 人のユーザがサーバ全体で別のユーザの ID を利用できます。「代理権限の使用」(561 ページ) を参照してください。

setuser の使用

データベース所有者は、次の場合に setuser を使用できます。

- 別のユーザが所有するオブジェクトにアクセスする場合。
- 別のユーザが所有するオブジェクトに対するパーミッションを付与する場合。
- 別のユーザが所有者となるオブジェクトを作成する場合。
- 何らかの理由で別のユーザの DAC パーミッションを一時的に利用する場合。

setuser コマンドを実行すると、データベース所有者は自動的に別のユーザの DAC パーミッションを取得できますが、このコマンドは既に付与されている役割には影響しません。

setuser パーミッションは、デフォルトではデータベース所有者に付与されており、譲渡することはできません。なり代わるユーザは、そのデータベースのアクセス権を持つユーザでなければなりません。Adaptive Server は、なり代わるユーザのパーミッションをチェックします。

システム管理者は、setuser を使用して、別のユーザが所有するオブジェクトを作成できます。ただし、システム管理者は、DAC パーミッション・システムの外部で操作するため、setuser を使用して別のユーザのパーミッションを取得する必要はありません。setuser コマンドは、次の setuser コマンドが実行されるか、現在のデータベースが変更されるか、あるいはユーザがログオフするまで有効です。

構文は次のとおりです。

```
setuser ["user_name"]
```

この *user_name* は、ID を使用される、データベース内の有効なユーザです。

元の ID に戻るには、*user_name* の値を指定しないで setuser コマンドを実行します。

次の例は、データベース所有者が、Mary が所有する authors テーブルを読み込むパーミッションを Joe に付与する方法を示します。

```
setuser "mary"
grant select on authors to joe
setuser     /*reestablishes original identity*/
```

代理権限の使用

Adaptive Server の代理権限機能を使用すると、システム・セキュリティ担当者は、別のユーザのセキュリティ・コンテキストを利用する機能を、選択したログインに付与できます。また、さまざまなユーザに代わってアプリケーションでタスクを実行する方法を制御できます。代理権限を使用するパーミッションを持つログインは Adaptive Server 内の別のログインになり代わることができます。

警告！ 他のユーザ ID を利用する機能は非常に強力なものであるため、信頼された管理者とアプリケーションだけに利用を限定する必要があります。grant set proxy ... restrict role を使用すると、ID を切り替えたときにユーザが特定の役割を取得できないように制限できます。

---

set proxy または set session authorization を実行するユーザは、被代理ユーザのログイン名とサーバ・ユーザ ID の両方を使用して操作を行います。ログイン名は、master..syslogins の name カラムに保管されています。また、サーバ・ユーザ ID は、master..syslogins の suid カラムに保管されています。これらの値は、サーバ全体のすべてのデータベース内でアクティブです。

注意 set proxy と set session authorization の機能は同じなので、どちらを使用してもかまいません。唯一の違いは、set session authorization が ANSI SQL92 互換であるのに対し、set proxy は Transact-SQL の拡張機能であるという点です。

set proxy を使用した役割の制限

set proxy...restricted role を付与することによって、ID を切り替えたときに特定の役割を取得できないように制限できます。

set proxy の構文は次のとおりです。

```
grant set proxy to user | role
    [restrict role role_list | all | system]
```

各パラメータの意味は、次のとおりです。

- *role_list* − ターゲット・ログインに対して制限する役割のリスト。付与対象者が、このリストのすべての役割を持っていることが必要です。そうでない場合は set proxy コマンドが失敗します。
- *all* − 付与対象者と同じ役割、またはその役割のサブセットを持つユーザについてのみ set proxy を実行できるようにします。
- *system* − 付与対象者がターゲット・ログインと同じシステム役割の組み合わせを持つようにします。

この例は、set proxy をユーザ “joe” に付与しますが、“joe” が ID を、sa、sso、または admin の役割を持つユーザに切り替えることは制限します (ただし、“joe” が既にこれらの役割を持っている場合は、これらの役割を持つユーザに対して set proxy を実行できます)。

```
grant set proxy to joe
restrict role sa_role, sso_role, admin_role
```

“joe” が admin_role を持つユーザ (この例では Our_admin_role) に ID を切り替えようとした場合、joe が admin_role を持っていないかぎ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 は、トランザクション内では実行できません。
- ロックされたログインを使用して、他のユーザの代理となることはできません。たとえば、“joseph” がロックされたログインの場合、次のコマンドは許可されません。

  ```
  set proxy "joseph"
  ```

- set proxy と set session authorization は、実行するユーザが使用許可を持つすべてのデータベースから実行できます。ただし、指定する *login_name* がデータベース内の有効なユーザであるか、データベースに “guest” が定義されている必要があります。
- 許可されるのは 1 レベルだけです。複数のユーザの代理権限を使用する場合は、それぞれの権限の使用を終了するたびに元の ID に戻る必要があります。
- set proxy または set session authorization をプロシージャ内から実行すると、プロシージャの終了時に自動的に元の ID に戻ります。

自分のログインに set proxy または set session authorization を使用するためのパーミッションが付与されている場合は、これらのコマンドを使用して、別のユーザになり代わることができます。構文は次のとおりです。*login_name* は、master..syslogins 内の有効なログイン名です。

set proxy *login_name*

または

set session authorization *login_name*

ログイン名は引用符で囲んでください。

たとえば、“mary” の代理権限を使用するには、次のコマンドを実行します。

```
set proxy "mary"
```

代理権限を設定したら、サーバでの自分のログイン名と、データベースでの自分のユーザ名を確認します。たとえば、自分のログインが “ralph” であり、set proxy 権限が付与されているものと想定します。このとき、データベース pubs2 において、“sallyn” および “ralph” としていくつかのコマンドを実行します。“sallyn” には、このデータベースでの有効な名前 (“sally”) がありますが、Ralph と Rudolph にはありません。ただし、pubs2 には “guest” ユーザが定義されています。そこで、次のコマンドを実行できます。

```
set proxy "sallyn"
go
use pubs2
go
select suser_name(), user_name()
go
------------------------------ --------------------
sallyn                         sally
```

Rudolph に変更するには、まず自身の ID に戻ります。これには、次のコマンドを実行します。

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

Ralph は、このデータベース内では “guest” であることに注意してください。

さらに、次のコマンドを実行します。

```
set proxy "rudolph"
go
select suser_name(), user_name()
go
------------------------------ --------------------
rudolph                        guest
```

Rudolph もデータベース内の有効なユーザではないため、このデータベースでは guest になっています。

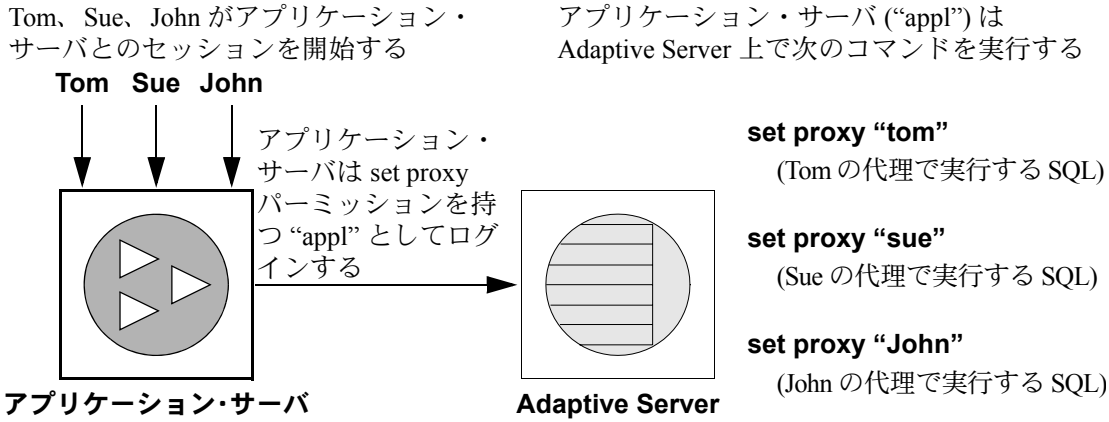
今度は、“sa” アカウントになり代わります。次のコマンドを実行します。

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

アプリケーションによる代理権限の使用方法

図 17-1 は、汎用ログイン “appl” を使用して Adaptive Server にログインし、多数のユーザに代わってプロシージャとコマンドを実行するアプリケーション・サーバを示します。“appl” が Tom になり代わっている間、アプリケーションは Tom のパーミッションを持ちます。同様に、“appl” が Sue と John になり代わると、アプリケーションは、それぞれ Sue と John のパーミッションだけを持ちます。

図 *17-1:* アプリケーションと代理権限



パーミッションを表示する方法

表 17-5 は、代理パーミッション、オブジェクト作成パーミッション、オブジェクト・アクセス・パーミッションに関する情報をレポートするためのシステム・プロシージャを示します。

表 *17-5:* パーミッションについてレポートするシステム・プロシージャ

レポートする情報	使用
代理	システム・テーブル
ユーザとプロセス	sp_who
データベース・オブジェクトまたはユーザに対するパーミッション	sp_helprotect
特定のテーブルに対するパーミッション	sp_table_privileges
テーブル内の特定のカラムに対するパーミッション	sp_column_privileges

代理権限に対する *sysprotects* テーブルの問い合わせ

ユーザ、グループ、役割に付与されているパーミッションや取り消されたパーミッションに関する情報を表示するには、sysprotects テーブルを問い合わせます。action カラムは、パーミッションを表します。たとえば、set proxy と set session authorization の action の値はどちらも 167 です。

たとえば、次のクエリを実行します。

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

このクエリを実行すると、パーミッションを付与または取り消したユーザのユーザ ID (grantor カラム)、パーミッションを持つユーザのユーザ ID (uid カラム)、保護のタイプ (protecttype カラム) が表示されます。protecttype カラムには、次の値が含まれます。

- grant with grant を示す 0
- grant を示す 1
- revoke を示す 2

sysprotects テーブルの詳細については、『リファレンス・マニュアル：ビルディング・ブロック』を参照してください。

ユーザとプロセスに関する情報の表示方法

sp_who は、現在のすべての Adaptive Server ユーザとプロセスに関する情報、または特定のユーザやプロセスに関する情報を表示します。sp_who の実行結果の中に、loginame と orignameがあります。ユーザが代理権限のもとで操作を行っている場合、origname には元のログイン名が表示されます。たとえば、“ralph” が次のコマンドを実行してから、いくつかの SQL コマンドを実行するとします。

```
set proxy susie
```

sp_who は、loginame として “susie” を返し、origname として “ralph” を返します。

sp_who は、masater..sysprocesses システム・テーブルを問い合わせます。このテーブルには、サーバ・ユーザ ID のカラム (suid) と元のサーバ・ユーザ ID のカラム (origsuid) があります。

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

データベース・オブジェクトまたはユーザに対するパーミッション

データベース・オブジェクトまたはユーザごとのパーミッションについて表示するには、sp_helprotect を使用します。指定したオブジェクトのユーザごとのパーミッションを表示することもできます。このプロシージャは、すべてのユーザが実行できます。構文は次のとおりです。

```
sp_helprotect [name [, username [, "grant"
    [,"none"|"granted"|"enabled"|role_name]]]]
```

各パラメータの意味は、次のとおりです。

- *name* は、テーブル、ビュー、またはストアド・プロシージャの名前、あるいは現在のデータベース内のユーザ、グループ、または役割の名前です。名前を指定しない場合、sp_helprotect はデータベース内のパーミッションすべてをレポートします。
- *username* は、現在のデータベース内のユーザの名前です。

  username を指定すると、指定したオブジェクトに対してそのユーザに付与されているパーミッションだけがレポートされます。*name* がオブジェクトではない場合は、sp_helprotect を実行すると *name* がユーザ、グループ、役割のどれに該当するかが検査され、これらのいずれかである場合は、そのユーザ、グループ、または役割に対するパーミッションが表示されます。キーワード grant を指定し、*name* にオブジェクト以外のものを指定して sp_helprotect を実行すると、with grant option によって付与されたすべてのパーミッションが表示されます。
- grant を指定すると、with grant option で *name* に付与されているパーミッションが表示されます。
- none を指定すると、ユーザに付与されている役割は無視されます。
- granted を指定すると、ユーザに付与されているすべての役割に関する情報も表示されます。
- enabled を指定すると、そのユーザがアクティブ化したすべての役割に関する情報も表示されます。
- *role_name* を指定すると、この役割がユーザに付与されているかどうかには関係なく、指定した役割に関するパーミッション情報だけが表示されます。

たとえば、次の一連の grant 文と revoke 文を発行するとします。

```
grant select on titles to judy
grant update on titles to judy
revoke update on titles(contract) from judy
grant select on publishers to judy
    with grant option
```

Judy が titles テーブルの各カラムに対して現在持っているパーミッションを調べるには、次のように入力します。

```
    sp_helprotect titles, judy
grantor grantee type   action   object     column      grantable
------- ------ -----    ------    ------      ------       -------
dbo     judy   Grant   Select   titles     All         FALSE
dbo     judy   Grant   Update   titles     advance     FALSE
dbo     judy   Grant   Update   titles     notes       FALSE
dbo     judy   Grant   Update   titles     price       FALSE
dbo     judy   Grant   Update   titles     pub_id      FALSE
dbo     judy   Grant   Update   titles     pubdate     FALSE
dbo     judy   Grant   Update   titles     title       FALSE
dbo     judy   Grant   Update   titles     title_id    FALSE
dbo     judy   Grant   Update   titles     total_sales FALSE
dbo     judy    Grant   Update   titles     type         FALSE
```

最初の行は、データベース所有者 (“dbo” が Judy に titles テーブルのすべてのカラムを選択できるパーミッションを付与していることを示します。残りの行は、Judy は表示されているカラムの更新だけができることを示しています。つまり、他のユーザに select パーミッションや update パーミッションを付与することはできません。

publishers テーブルに対する Judy のパーミッションを調べるには、次のように入力します。

```
    sp_helprotect publishers, judy
```

次の表示では、grantable カラムの値が TRUE です。つまり、Judy は他のユーザにパーミッションを付与できます。

```
grantor grantee type   action   object     column      grantable
------- ------ -----    ------    ------      ------       -------
dbo     judy    Grant   Select  publishers   all          TRUE
```

特定のテーブルに対するパーミッションを表示する方法

指定したテーブルに関するパーミッション情報を表示するには、sp_table_privileges を使用します。構文は次のとおりです。

```
sp_table_privileges table_name [, table_owner
    [, table_qualifier]]
```

各パラメータの意味は、次のとおりです。

- *table_name* テーブルの名前です。これは必須です。
- *table_owner* は、テーブル所有者が “dbo” でも sp_table_privileges を実行するユーザでもない場合に、テーブル所有者の名前を指定するのに使用します。
- *table_qualifier* は現在のデータベースの名前です。

省略するパラメータには、null を使用します。

たとえば、次の文は、titles テーブルについて付与されているすべてのパーミッションの情報を返します。

```
sp_table_privileges titles
```

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

特定のカラムに対するパーミッションを表示する方法

テーブル内のカラムに対するパーミッションに関する情報を表示するには、sp_column_privileges を使用します。構文は次のとおりです。

```
sp_column_privileges table_name [, table_owner
    [, table_qualifier [, column_name]]]
```

各パラメータの意味は、次のとおりです。

- *table_name* はテーブルの名前です。
- *table_owner* は、テーブル所有者が “dbo” でも sp_column_privileges を実行するユーザでもない場合に、テーブル所有者の名前を指定するのに使用します。
- *table_qualifier* は現在のデータベースの名前です。
- *column_name* には、パーミッション情報を表示するカラムの名前を指定します。

省略するパラメータには、null を使用します。

たとえば、次の文は、publishers テーブルの pub_id カラムについての情報を返します。

```
sp_column_privileges publishers, null, null, pub_id
```

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

セキュリティ・メカニズムとしてのビューとストアド・プロシージャの使用

ビューとストアド・プロシージャは、セキュリティ・メカニズムとして使用できます。ビューやストアド・プロシージャを使用することにより、ユーザがデータに直接アクセスできないようにして、データベース・オブジェクトへのユーザのアクセスを制御することができます。たとえば、projects テーブル内のコスト情報を更新するプロシージャに対する execute パーミッションを担当者に付与すれば、そのテーブル内の機密データをユーザが参照できないようにすることができます。この機能を活用するには、ビューやストアド・プロシージャを作成するユーザが、そのプロシージャやビューだけでなく、基本となるオブジェクトも所有する必要があります。基本となるオブジェクトを所有していない場合は、ビューやストアド・プロシージャを利用するユーザが、そのオブジェクトにアクセスするためのパーミッションを持っていなければなりません。パーミッションが必要な場合の詳細については、「所有権の連鎖の理解」(573 ページ) を参照してください。

ビューまたはプロシージャを使用するとき、必要に応じて Adaptive Server によるパーミッションの検査が行われます。ビューまたはプロシージャを作成するときには、基本となるオブジェクトに対するパーミッション検査は行われません。

セキュリティ・メカニズムとしてのビューの使用

ビューを使用して表示できるデータに対してだけ、問い合わせや変更ができます。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。

たとえば、一部のユーザを、金銭と売上に関する titles テーブル内のカラムにアクセスできないようにしたいと仮定します。その場合には、金銭と売上に関するカラムを除いて titles テーブルのビューを作成し、そのビューに対するパーミッションをすべてのユーザに付与して、テーブルに対するパーミッションは営業部門にだけ付与します。

```
grant all on bookview to public
grant all on titles to sales
```

これらの権限の条件をビューを使わずに設定するには、次の文を使用します。

```
grant all on titles to public
revoke select, update on titles (price, advance,
    total_sales)
from public
grant select, update on titles (price, advance,
    total_sales)
to sales
```

この 2 番目の方法を使用した場合は、sales グループのメンバでないユーザが select * from titles コマンドを入力したときに、次の語句が含まれるメッセージが突然表示されて混乱を招くおそれがあります。

```
パーミッションが拒否されました
```

このアスタリスクは、titles テーブル内のすべてのカラムのリストに展開されます。このリスト内のカラムのいくつかについては、営業部門以外のユーザからはパーミッションが取り消されているので、そのカラムに対するアクセスは拒否されます。ユーザがアクセス権を持っていないカラムがエラー・メッセージに表示されます。

営業部門以外のユーザが、パーミッションを持つすべてのカラムを表示するには、カラムを明示的に指定する必要があります。このため、ビューを作成して、適切なパーミッションを付与する方が簡単です。

ビューを使用すると、「コンテキストで区別されるプロテクション」を実現することもできます。たとえば、データ入力者に、自分が追加または更新したローだけにアクセスできるパーミッションを付与するビューを作成します。これを行うには、テーブルにカラムを追加し、各ローを入力したユーザのユーザ ID が自動的にデフォルト値によってそのカラムに記録されるようにします。このデフォルト値は、create table 文で次のように定義します。

```
create table testtable
    (empid       int,
     startdate   datetime,
     username     varchar(30) default user)
```

次に、このテーブルのローのうち、uid が現在のユーザに等しいローがすべて表示されるビューを定義します。

```
create view context_view
as
    select *
    from testtable
    where username = user_name()
with check option
```

このビューによって検索できるローは、ビューに対して select コマンドを発行するユーザの ID によって異なります。ビュー定義に with check option を追加すると、データ入力者が username カラム内の情報を改ざんできないようにすることができます。

セキュリティ・メカニズムとしてのストアド・プロシージャの使用

ストアド・プロシージャの所有者と、基本となるすべてのオブジェクトの所有者が同じならば、プロシージャを使うためのパーミッションを所有者が他のユーザに付与するときに、基本となるオブジェクトに対するパーミッションを付与する必要はありません。たとえば、指定されたテーブルのローとカラムのサブセットを更新するストアド・プロシージャを実行するためのパーミッションをユーザに付与するとき、ユーザはそのテーブルに対するその他のパーミッションを持っていなくてもかまいません。

役割とストアド・プロシージャ

grant execute コマンドを使うと、ストアド・プロシージャに対する実行パーミッションを、指定した役割を付与されているすべてのユーザに付与できます。同様に revoke execute コマンドを使って、このパーミッションを削除できます。ただし、grant execute パーミッションによる方法では、特定の役割を持たないユーザにストアド・プロシージャの実行パーミッションが付与されることを防ぐことはできません。

セキュリティをさらに高めるには、プロシージャ内で proc_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 proc_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、proc_role を使用して、ユーザがシステム管理者の役割を持っているかどうかを確認するプロシージャを次に示します。

```
create proc test_proc
as
if (proc_role("sa_role") = 0)
begin
    print "You don't have the right role"
    return -1
end
else
    print "You have SA role"
    return 0
```

proc_role の詳細については、『リファレンス・マニュアル：ビルディング・ブロック』の「第 2 章 Transact-SQL 関数」の「システム関数」を参照してください。

所有権の連鎖の理解

ビューは別のビューやテーブルに従属します。プロシージャは別のプロシージャ、ビュー、またはテーブルに従属します。このような従属性を「所有権の連鎖」と考えることができます。

通常は、ビューの所有者はその基本となるオブジェクト (他のビューやテーブル) も所有します。ストアド・プロシージャの所有者は、そのプロシージャによって参照されるすべてのプロシージャ、テーブル、ビューを所有します。

ビューとその基本となるオブジェクトは、ストアド・プロシージャとそれが参照するすべてのオブジェクトと同様に、通常はすべて同じデータベース内に存在しますが、これは必須ではありません。これらのオブジェクトが別のデータベース内に存在する場合は、ビューまたはストアド・プロシージャを使用するユーザは、オブジェクトが存在するすべてのデータベース内の有効なユーザか guest ユーザである必要があります。このため、データベース所有者による許可を受けなければ、ユーザはデータベースにアクセスできません。

プロシージャまたはビューに対する execute パーミッションを付与されているユーザがそのプロシージャまたはビューを使用するときに、次の条件に該当する場合は、基本となるオブジェクトのパーミッションの検査は一切行われません。

- これらのオブジェクトとビューまたはプロシージャが同じユーザによって所有されている場合。
- ビューまたはプロシージャにアクセスするユーザが、基本となるオブジェクトが存在するそれぞれのデータベース内の有効なユーザか guest ユーザである場合。

ただし、すべてのオブジェクトの所有者が同じでない場合は、所有権の連鎖が切れたところでオブジェクトのパーミッションの検査が行われます。つまり、オブジェクト A がオブジェクト B を参照していて、オブジェクト A の所有者とオブジェクト B の所有者が異なる場合は、オブジェクト B に対するパーミッションが検査されます。このようにして、データへのアクセスをどのユーザに許可するかという制御を元のデータの所有者が維持できるようにします。

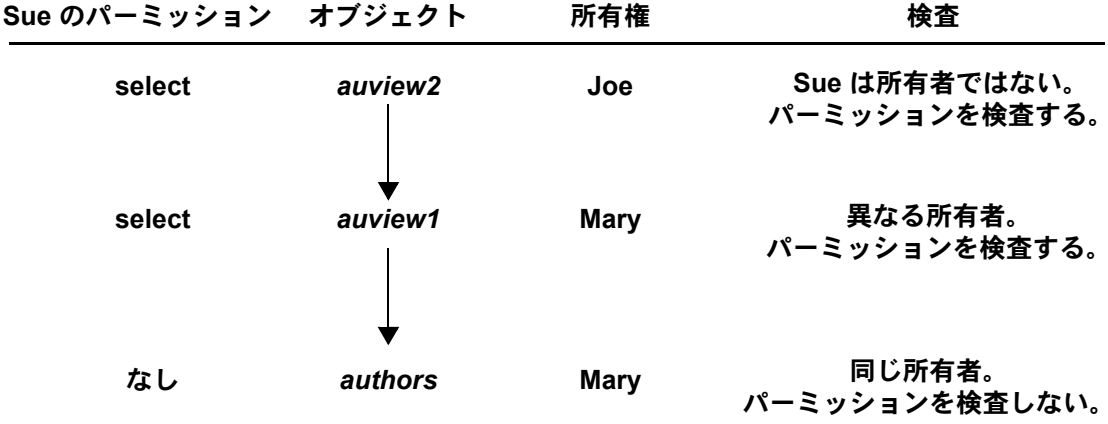
通常は、ビューを作成するユーザが注意しなければならないのは、そのビューに対するパーミッションの付与だけです。たとえば、Mary が、自分が所有する authors テーブルに auview1 というビューを作成したとします。Mary が auview1 に対する select パーミッションを Sue に付与すると、Sue がこのビューにアクセスするとき、authors に対するパーミッションの検査は行われません。

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

ビューと所有権の連鎖の例

Joe が作成する auview2 というビューが、Mary のビュー auview1 に従属するとします。Joe は auview2 に対する select パーミッションを Sue に付与します。

図 *17-2:* ビューの所有権の連鎖とパーミッション検査 *(ケース 1)*

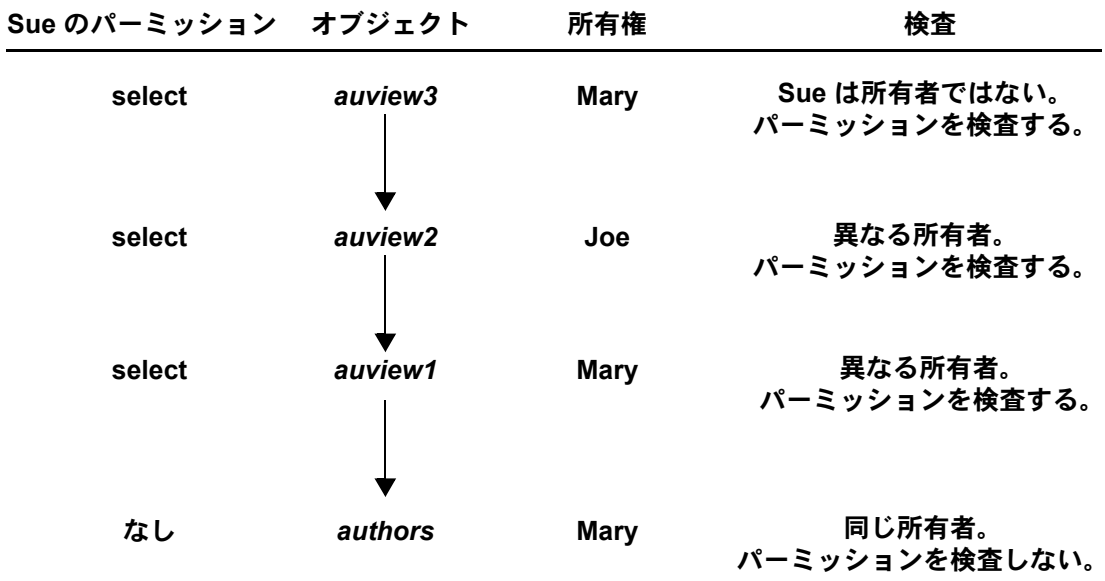


Adaptive Server は auview2 と auview1 に対するパーミッションを検査して、Sue がこれらのビューを使用できると判断します。また、auview1 と authors に対する所有権を検査して、これらの所有者が同じであると判断します。したがって、Sue は auview2 を使用できます。

この例をさらに一歩進めて、Joe のビュー auview2 が auview1 に従属していて、auview1 が authors に従属しているとします。Mary は、Joe の auview2 の上に auview3 を作成します。auview1 と authors は Mary によって所有されます。

所有権の連鎖は次のようになります。

図 *17-3:* ビューの所有権の連鎖とパーミッション検査 *(ケース 2)*

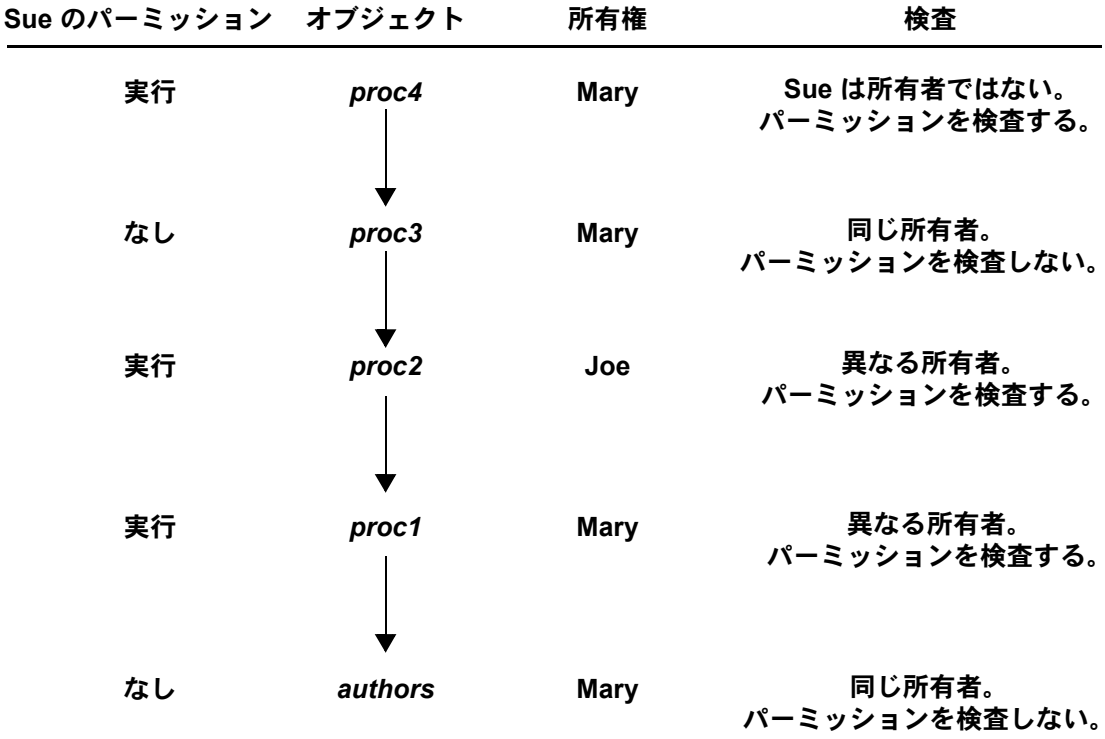


Sue が auview3 にアクセスすると、Adaptive Server は、auview3、auview2、auview1 に対するパーミッションを検査します。auview2 に対するパーミッションが Joe から Sue に付与され、auview3 と auview1 に対するパーミッションが Mary から付与されていれば、Adaptive Server はアクセスを許可します。Adaptive Server によってパーミッションの検査が行われるのは、連鎖内の直前のオブジェクトが別の所有者によって所有されている場合 (またはそのオブジェクトが連鎖内の最初のオブジェクトである場合) だけです。たとえば、auview2 は検査の対象です。これは、直前のオブジェクト auview3 が別のユーザによって所有されているからです。authors に対するパーミッションは検査されません。authors に直接従属しているオブジェクト auview1 が同じユーザによって所有されているからです。

プロシージャと所有権の連鎖の例

プロシージャはビューと同じ規則に従います。たとえば、所有権の連鎖が次のようになっているとします。

図 *17-4:* ストアド・プロシージャの所有権の連鎖とパーミッション検査



Sue が proc4 を実行するには、proc4、proc2、proc1 を実行するためのパーミッションが必要です。proc3 は proc4 と所有者が同じなので、proc3 を実行するためのパーミッションは必要ありません。

Adaptive Server は、Sue が proc4 を実行するたびに、proc4 とこのプロシージャが参照するすべてのオブジェクトに対する Sue のパーミッションを検査します。Adaptive Server は、参照されるオブジェクトのうちどれを検査するかを把握しています。この情報は Sue が proc4 を初めて実行したときに決定され、プロシージャの実行プランとともに保管されています。プロシージャによって参照されるオブジェクトが削除されたり再定義されたりしないかぎり、検査するオブジェクトについての最初の決定は変更されません。

この保護階層を使用すれば、オブジェクトの所有者がそのオブジェクトに対するアクセスを完全に制御できます。所有者は、テーブルへのアクセスだけではなく、ビューやストアド・プロシージャへのアクセスも制御できます。

トリガのパーミッション

「トリガ」は、整合性、特に参照整合性を保つために使用される特別な種類のストアド・プロシージャです。トリガは直接実行されることはなく、テーブルの変更の結果として実行されます。トリガに対するパーミッションを付与 (grant) または取り消す (revoke) 方法はありません。

オブジェクトに対してトリガを作成できるのは、そのオブジェクトの所有者だけです。ただし、テーブルに対するトリガが、別のユーザによって所有されているオブジェクトを参照する場合は、所有権の連鎖が切れることになります。プロシージャに適用される保護階層規則はトリガに対しても適用されます。

トリガが影響を与えるオブジェクトは、通常はそのトリガを所有するユーザが所有するオブジェクトですが、別のユーザが所有するオブジェクトを変更するトリガを作成することもできます。この場合は、トリガをアクティブにする方法でオブジェクトを変更するユーザはすべて、他のオブジェクトに対するパーミッションも持っている必要があります。

トリガが影響を与えるオブジェクトに対するパーミッションがユーザに付与されていないという理由で Adaptive Server がデータ変更コマンドに対するパーミッションを拒否した場合は、データ変更トランザクション全体がロールバックされます。

詳細については、『Transact-SQL ユーザーズ・ガイド』の「第 20 章 トリガ：参照整合性の保持」を参照してください。

ロー・レベル・アクセス制御の使用

ロー・レベル・アクセス制御には次の機能があり、データベース所有者やテーブル所有者は安全なデータ・アクセス環境を自動的に作成できます。

- より細密なデータ・セキュリティ。テーブルとカラムだけでなく、個々のローに対してパーミッションを設定できます。
- グループ、役割、アプリケーションに応じた自動データ・フィルタリング。
- サーバでのコード化によるデータ・レベルのセキュリティ。

ロー・レベル・アクセス制御の次の 3 つの機能によって、テーブルの個々のローのデータへのアクセスを制御します。

- データベース所有者が定義してテーブルにバインドするアクセス・ルール。
- ユーザ定義のコンテキストを定義、保存、検索するための組み込み関数の集合である Application Context Facility。
- データベース所有者、sa_role、またはユーザが作成できるログイン・トリガ。

Adaptive Server のロー・レベル・アクセス制御はすべてのデータ操作言語 (DML) に適用されるので、ユーザがアクセス制御を回避してデータを取得することはできません。

ロー・レベル・アクセス制御を使用するようにシステムを設定する構文は次のとおりです。

```
sp_configure "enable row level access", 1
```

このオプションを使用するときは、Adaptive Server が使用するメモリの量がわずかに増えます。また、ASE_RLAC ライセンス・オプションが必要です。ロー・レベル・アクセス制御は動的オプションなので、Adaptive Server を再起動する必要はありません。

アクセス・ルール

ロー・レベル・アクセス制御機能を使用するには、既存の create rule の構文に access オプションを追加します。アクセス・ルールは、参照または変更できるローを制限するものです。

アクセス・ルールは、特定のカラムでユーザが挿入または更新できる値をテーブル所有者が制御するためのドメイン・ルールに似ています。ドメイン・ルールは、追加されるデータについて制限を適用するもので、update コマンドと insert コマンドに対して機能します。

アクセス・ルールは、検索されるデータを制限するもので、select、update、delete の各オペレーションに適用されます。アクセス・ルールは、クエリで読み込まれるすべてのカラムに対して適用されます。select リストで指定されていないカラムについても同様です。つまり、特定のクエリにおいて、更新されるテーブルにはドメイン・ルールが適用され、読み込まれるすべてのテーブルにアクセス・ルールが適用されます。

次に例を示します。

```
insert into orders_table
select * from old_orders_table
```

このクエリでは、orders_table に対するドメイン・ルールと old_orders_table に対するアクセス・ルールがある場合に、orders_table は更新されるのでドメイン・ルールが適用され、old_orders_table は読み取られるのでアクセス・ルールが適用されます。

アクセス・ルールを使用することは、ビューを使用することや、where 句のあるアドホック・クエリを使用することに似ています。アクセス・ルールが付加された後でクエリのコンパイルと最適化が行われるので、パフォーマンスが低下することはありません。アクセス・ルールは、テーブル・データの仮想ビューを実現するものです。つまり、カラムにバインドされた特定のアクセス・ルールに応じて変化するビューです。

アクセス・ルールは、sp_addtype を使用して定義するユーザ定義データ型にバインドできます。アクセス・ルールはユーザ・テーブルに対して適用されます。これを利用すれば、テーブル所有者やデータベース所有者が正規化スキーマの中でカラムにアクセス・ルールをバインドするという管理作業を行う必要はありません。たとえば、ベース型が varchar(30) であるユーザ定義データ型を作成して username という名前を付け、このデータ型にアクセス・ルールをバインドしたとします。このアクセス・ルールは、アプリケーション内の username 型のカラムを持つすべてのテーブルに適用されます。

アプリケーション開発者は、Java とアプリケーション・コンテキストを使って柔軟なアクセス・ルールを作成できます。詳細については、「ユーザ定義 Java 関数としてのアクセス・ルール」(584 ページ) と「Application Context Facility の使用」(587 ページ) を参照してください。

アクセス・ルールの構文

アクセス・ルールを作成するには、create rule 構文の access パラメータを使用します。

```
create [or|and] access rule (access_rule_name)
as (condition)
```

アクセス・ルールを持つサンプル・テーブルの作成

この項では、テーブルを作成してアクセス・ルールをバインドするプロセスを示します。

テーブルの作成

テーブル所有者は、テーブル T を作成して (username char(30)、title char(30)、classified_data char(1024))、次のデータを入力します。

```
AA, "Administrative Assistant","Memo to President"
AA, "Administrative Assistant","Tracking Stock Movements"
VP1, "Vice President", "Meeting Schedule"
VP2, "Vice President", "Meeting Schedule"
```

アクセス・ルールの作成とバインド

テーブル所有者は、アクセス・ルール uname_acc_rule を作成して、テーブル T の username カラムにバインドします。

```
create access rule uname_acc_rule
as @username = suser_name()
-----------
sp_bindrule uname_acc_rule, "T.username"
```

テーブルに対するクエリ

次のクエリを発行します。

```
select * from T
```

Adaptive Server は、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・ルールの使用

アクセス・ルールの作成

次の手順で、アクセス・ルールを作成します。

```
create access rule empid1_access
as @empid = 1

create access rule deptno1_access
as @deptid = 2
```

次の手順で、OR アクセス・ルールを作成します。

```
create or access rule name1_access
as @name = "smith"

create or access rule phone_access
as @phone = "9999"
```

テーブルの作成

次の手順で、テスト・テーブルを作成します。

```
create table testtab1 (empno int, deptno int,name char(10),
phone char(4))
```

テーブルへのルールのバインド

次の手順で、テスト・テーブルのカラムにアクセス・ルールをバインドします。

```
sp_bindrule empid1_access, "testtab1.empno"
/*Rule bound to table column.*/
(return status = 0)

sp_bindrule deptno1_access,"testtab1.deptno"
/*Rule bound to table column.*/

(return status = 0)

sp_bindrule name1_access,"testtab1.name"
/*Rule bound to table column.*/

(return status = 0)

sp_bindrule phone_access,"testtab1.phone"
/*Rule bound to table column.*/

(return status = 0)
```

テーブルへのデータの挿入

次の手順で、テスト・テーブルに値を挿入します。

```
insert testtab1 values (1,1,"smith","3245")
(1 row affected)

insert testtab1 values(2,1,"jones","0283")
(1 row affected)

insert testtab1 values(1,2,"smith","8282")(1 row affected)

insert testtab1 values (2,2,"smith","9999")

(1 row affected)
```

アクセス・ルールの例

次の例では、アクセス・ルールによって返されるローの内容が、どのようにアクセス・ルールによって制限されているかを示します。

例 1

この例では、2 つのローの情報が返されます。

```
/* 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　この例では、4 つのローの情報が返されます。

```
/* 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　この例では、6 つのローの情報が返されます。

```
/* return the rows when name = "smith" or phone = "9999" */

select * from testtab1

 empno       deptno      name       phone
 ----------- ----------- ---------- -----
           1           1 smith      3245
           1           2 smith      8282
           2           2 smith      9999
           3           2 smith      8888
           1           2 jones      9999
           2           3 jones      9999
```

アクセス・ルールと alter table コマンド

テーブル所有者が alter table コマンドを実行するとき、コマンド実行中はアクセス・ルールは無効になり、コマンド実行終了時に再び有効化されます。アクセス・ルールが無効化されるのは、alter table コマンドの実行中にテーブル・データをフィルタしないようにするためです。

アクセス・ルールと *bcp* コマンド

bcp を使ってテーブルからデータをコピーするときは、アクセス・ルールが適用されます。alter table の場合とは異なり、Adaptive Server がアクセス・ルールを無効にすることはできません。これは、bcp はテーブルに対する選択パーミッションを持つユーザであれば誰でも使用できるためです。

セキュリティのために、データベース所有者は、バルク・コピー・アウトの実行中はテーブルを排他的にロックし、アクセス・ルールを無効にします。アクセス・ルールが無効化されている間は、ロックによって他のユーザのアクセスを不可能にします。データベース所有者は、データのコピーが完了したら、アクセス・ルールをバインドし、テーブルのロックを解除します。

ユーザ定義 Java 関数としてのアクセス・ルール

アクセス・ルールでは、ユーザ定義 Java 関数を使用できます。たとえば、アプリケーションのプロファイル、アプリケーションにログインしたユーザ、アプリケーションを実行するために現在ユーザに与えられている役割などを使用する高度なルールを作成する場合に、Java 関数を使用します。

次の GetSecVal メソッドを使用する Java クラスでは、JDBC を使用する Java メソッドをユーザ定義関数としてアクセス･ルール内で使用する方法を示します。

```
import java.sql.*;
import java.util.*;

public class sec_class {
static String _url = "jdbc:sybase:asejdbc";
public static int GetSecVal(int c1)
{
try
{
PreparedStatement pstmt;
ResultSet rs = null;
Connection con = null;
    int pno_val;

pstmt = null;

Class.forName("sybase.asejdbc.ASEDriver");
con = DriverManager.getConnection(_url);

if (con == null)
{
return (-1);
}

pstmt = con.prepareStatement("select classification from
sec_tab where id = ?");

if (pstmt == null)
```

```
{
return (-1);
}

pstmt.setInt(1, c1);

rs = pstmt.executeQuery();

rs.next();

pno_val = rs.getInt(1);

rs.close();

pstmt.close();

con.close();

return (pno_val);

}
catch (SQLException sqe)
{
return(sqe.getErrorCode());
}
catch (ClassNotFoundException e)
{

System.out.println("Unexpected exception :" + e.toString());
System.out.println("¥nThis error usually indicates that " +
"your Java CLASSPATH environment has not been set properly.");
e.printStackTrace();


return (-1);
}
catch (Exception e)
{
System.out.println("Unexpected exception :" + e.toString());
e.printStackTrace();
return (-1);
}
}
}
```

次のように、この Java コードのコンパイル後、同じプログラムを isql から実行できます。

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 から構成されます。ユーザは、各コンテキストに対してコンテキスト名、属性、値を定義します。Sybase が提供するデフォルトの読み込み専用アプリケーション・コンテキスト SYS_SESSION を使用すると、セッション固有の情報にアクセスできます。このアプリケーション・コンテキストの説明は、表 17-6 (594 ページ) を参照してください。また、「アプリケーション・コンテキストの作成と使用」(589 ページ) で説明しているように、ユーザが独自のアプリケーション・コンテキストを作成することもできます。

ユーザ・プロファイルとアプリケーション・プロファイル (システム管理者が作成するテーブルで定義される) を組み合わせることにより、複数のセキュリティ方式の累積や重ね合わせが可能となります。

ACF を使用すると、ユーザは次のものを定義、保存、検索できます。

- ユーザ・プロファイル (ユーザに付与された役割、およびユーザが属するグループ)
- 現在使用されているアプリケーション・プロファイル

1 つのセッションで使用できるアプリケーション・コンテキストの数に制限はありません。また、1 つのコンテキストで定義できる属性と値のペアの数も制限はありません。ACF コンテキストのローは 1 つのセッションに固有であり、複数のセッションにわたっては存続しません。ただし、ローカル変数とは異なり、ネストした文が実行されるときも、レベルを越えて利用可能です。ACF は、このようなコンテキスト・ローを設定、取得、検索、削除する組み込み関数の集まりです。

アプリケーション・コンテキスト関数を使ってパーミッションを設定する

アプリケーション・コンテキスト関数は、select 文の中で実行します。関数の所有者はサーバのシステム管理者です。アプリケーション・コンテキストを作成、設定、検索、削除するには、組み込み関数を使用します。

この関数で使用されるデータは、全テーブルに対する全ログインのデータを含むテーブルで定義されます。このテーブルは、システム管理者によって作成されます。このテーブルの詳細については、「ログイン・トリガの使用」(595 ページ) を参照してください。

- set_appcontext() は保存を実行します。

```
select set_appcontext ("titles", "rlac", "1")
```

- get_appcontext() に、セッション内のコンテキストの 2 つの要素を渡すと、3 つ目の要素が返されます。

```
select get_appcontext ("titles", "rlac")
------------------------
1
```

これらの関数および list_appcontext と rm_appcontext の詳細については、「アプリケーション・コンテキストの作成と使用」(589 ページ) を参照してください。

権限の付与と取り消し

特定のデータベース内のオブジェクトに対するアクセス権限を、ユーザ、役割、グループに付与したり取り消したりすることができます。ただし、create database、set session authorization および connect のみは例外です。これらの権限を付与されるユーザは master データベースの有効なユーザでなければなりません。他の権限を使用するには、そのオブジェクトが存在するデータベースの有効なユーザでなければなりません。

関数を使用するということは、特別な処置をとらない限り、ログインしたユーザがそのセッションのプロファイルを再設定できてしまうということです。Adaptive Server は組み込み関数を監査しますが、問題に気づく前にセキュリティが損なわれている可能性もあります。これらの組み込み関数へのアクセスを制限するには、権限の grant と revoke を使用します。sa_role を付与されたユーザだけが、組み込み関数に対する権限の付与と取り消しを実行できます。関数によって実行される、サーバによる強制データ・アクセス・コントロール・チェックの中では、select 権限のみがチェックされます。

有効なユーザ

関数にはオブジェクト ID はなく、ホーム・データベースもありません。したがって、各データベースの所有者は、関数に対する select 権限を該当するユーザに付与する必要があります。Adaptive Server は、ユーザのデフォルト・データベースを特定して、そのデータベースに対するパーミッションをチェックします。この方法では、データベース所有者による select 権限の付与が必要となるのはユーザのデフォルト・データベースだけです。他のデータベースについても制限が必要な場合は、そのデータベースの所有者が明示的にそのデータベースでのユーザの権限を取り消す必要があります。

関数に対する権限の付与や取り消しを行うときに、ユーザのデータ・アクセス制御チェックが行われるのは、アプリケーション・コンテキスト組み込み関数だけです。他の関数への権限の付与や取り消しを行っても Adaptive Server には何の影響も与えません。

public に付与されている権限の影響を受けるのは、システム管理者が作成するテーブルで指定されたユーザだけです。このテーブルの詳細については、「ログイン・トリガの使用」(595 ページ) を参照してください。guest ユーザが権限を持つのは、sa_role がこのテーブルに追加することによって明示的に権限を与えた場合だけです。

システム管理者は、以下のコマンドを実行して、特定のアプリケーション・コンテキスト関数に対する select 権限を付与または取り消します。

```
grant select on set_appcontext to user_role

grant select on set_appcontext to joe_user

revoke select on set_appcontext from joe_user
```

アプリケーション・コンテキストの作成と使用

アプリケーション・コンテキストの作成と管理に利用できる関数は以下のとおりです。詳細については、『リファレンス・マニュアル：ビルディング・ブロック』を参照してください。

- set_appcontext
- get_appcontext
- list_appcontext
- rm_appcontext

set_appcontext

指定されたユーザ・セッションのアプリケーション・コンテキスト名、属性名、属性値を設定します。これらは、アプリケーションの属性によって定義されます。

```
set_appcontext ("context_name", "attribute_name", "attribute_value")
```

- *context_name* － アプリケーション・コンテキスト名を指定するロー。データ型 char(30) として保存されます。
- *attribute_name* － アプリケーション・コンテキスト属性名を指定するロー。データ型 char(30) として保存されます。
- *attribute_value* アプリケーション属性値を指定するロー。データ型 char(255) として保存されます。

例

この例では、CONTEXT1 という名前のアプリケーション・コンテキストを作成し、属性 ATTR1 とその値 VALUE1 を設定します。

```
select set_appcontext("CONTEXT1", "ATTR1", "VALUE1")
---------------
0
```

この例では、既存のアプリケーション・コンテキストの上書きを試みます。試みは失敗し、-1 が返されます。

```
select set_appcontext("CONTEXT1", "ATTR1", "VALUE1")
--------------
-1
```

この例では、set_appcontext に値のデータ型の変換を組み込む方法を示します。

```
declare@val numeric
select @val = 20
select set_appcontext ("CONTEXT1", "ATTR2",
convert(char(20), @val))
------------
0
```

この例では、適切なパーミッションを持たないユーザがアプリケーション・コンテキストを設定しようとしたときの結果を示します。試みは失敗し、-1 が返されます。

```
select set_appcontext("CONTEXT1", "ATTR2", "VALUE1")
--------------
-1
```

使用法

- set_appcontext は、成功すると 0 を返し、失敗すると -1 を返します。
- 現在のセッションに既に存在する値を設定すると、set_appcontext は -1 を返します。
- set_appcontext では、既存のアプリケーション・コンテキストの値は上書きできません。コンテキストに新しい値を割り当てるには、コンテキストを削除してから、新しい値を使用して再作成してください。
- set_appcontext は、属性を char データ型として格納します。作成するアクセス・ルールで属性値を別のデータ型と比較する必要がある場合は、アクセス・ルールで char データを適切なデータ型に変換する必要があります。
- この関数では、すべての引数が必須です。

get_appcontext

指定されたコンテキストの属性値を返します。

get_appcontext ("*context_name*", "*attribute_name*")

- *context_name* − アプリケーション・コンテキスト名を指定するロー。データ型 char(30) として保存されます。
- *attribute_name* − アプリケーション・コンテキスト属性名を指定するロー。データ型 char(30) として保存されます。

例

この例では、ATTR1 に対して VALUE1 が返されます。

```
select get_appcontext ("CONTEXT1", "ATTR1")
-----------
VALUE1
```

ATTR1 は CONTEXT2 にはありません。

```
select get_appcontext("CONTEXT2", "ATTR1")
------------
NULL
```

この例では、適切なパーミッションを持たないユーザがアプリケーション・コンテキストを取得しようとしたときの結果を示します。

```
select get_appcontext("CONTEXT1", "ATTR2")
select permisssion denied on built-in get_appcontext, database
dbid
----------
-1
```

使用法

- get_appcontext は、成功すると 0 を返し、失敗すると -1 を返します。
- 指定された属性がアプリケーション・コンテキスト内にない場合は、get_appcontext は “null” を返します。
- get_appcontext は、属性を char データ型として格納します。作成するアクセス・ルールで、属性値を他のデータ型と比較する場合は、アクセス・ルールで char データを適切なデータ型に変換する必要があります。
- この関数では、すべての引数が必須です。

list_appcontext

現在のセッション内にある全コンテキストの属性をすべてリストします。

list_appcontext ("*context_name*")

- *context_name* − セッション内のアプリケーション・コンテキスト属性をすべて指定します。list_appcontext のデータ型は char(30) です。

例

list_appcontext を使用するには、ユーザに適切なパーミッションが付与されている必要があります。詳細については、「アプリケーション・コンテキスト関数を使ってパーミッションを設定する」(587 ページ) を参照してください。

この例では、適切なパーミッションを持つユーザがアプリケーション・コンテキストの一覧を表示したときの結果を示します。

```
select list_appcontext ("*", "*")
Context Name: (CONTEXT1)
Attribute Name: (ATTR1) Value: (VALUE2)
CHAPTER 17 Managing User Permissions
System Administration Guide: Volume 1 619
Context Name: (CONTEXT2)
Attribute Name: (ATTR1) Value: (VALUE!)
-----------
0
```

この例では、適切なパーミッションを持たないユーザがアプリケーション・コンテキストの一覧を表示しようとしたときの結果を示します。試みは失敗し、-1 が返されます。

```
select list_appcontext()
Select permission denied on built-in
list_appcontext, database DBID
---------
-1
```

使用法

- list_appcontext は、成功すると 0 を返し、失敗すると -1 を返します。
- 組み込み関数が複数の結果セットを返すことはないので、クライアント・アプリケーションは list_appcontext の戻り値をメッセージとして受け取ります。

rm_appcontext

特定のアプリケーション・コンテキストまたはすべてのアプリケーション・コンテキストを削除します。

rm_appcontext ("*context_name*", "*attribute_name*")

- *context_name* − アプリケーション・コンテキスト名を指定するロー。データ型 char(30) として保存されます。
- *attribute_name* − アプリケーション・コンテキスト属性名を指定するロー。データ型 char(30) として保存されます。

例

次の 3 つの例では、一部またはすべての属性を指定してアプリケーション・コンテキストを削除する方法を示します。アスタリスク ("*") を使用して、指定したコンテキスト内のすべての属性を削除します。

```
select rm_appcontext("CONTEXT1", "*")
---------
0
```

アスタリスク ("*") を使用して、すべてのコンテキストと属性を削除します。

```
select rm_appcontext("*", "*")
---------
0
```

この例では、存在しないコンテキストを削除しようとしています。試みは失敗し、-1 が返されます。

```
select rm_appcontext("NON_EXISTING_CTX", "ATTR2")
---------
-1
```

この例では、適切なパーミッションを持たないユーザがアプリケーション・コンテキストを削除しようとしたときの結果を示します。

```
select rm_appcontext("CONTEXT1", "ATTR2")
---------
-1
```

使用法

- rm_appcontext は、成功すると 0 を返し、失敗すると -1 を返します。
- この関数では、すべての引数が必須です。

SYS_SESSION システム・アプリケーション・コンテキスト

SYS_SESSION コンテキストを使用すると、デフォルトの事前定義アプリケーション・コンテキストが表示されます。これには、セッション固有の属性と値のペアが定義されています。このコンテキストを使用する構文は次のとおりです。

```
select list_appcontext ("SYS_SESSION", "*")
```

その後で、次の構文を使用します。

```
select get_appcontext ("SYS_SESSION", "<attribute>")
```

表 *17-6: SYS_SESSION* の属性と値

属性	値
username	ログイン名
hostname	クライアントの接続元ホスト名
applname	クライアントによって設定されたアプリケーション名
suserid	現在のデータベースでのユーザのユーザ ID
groupid	現在のデータベースでのユーザのグループ ID
dbid	ユーザの現在のデータベースの ID
dbname	現在のデータベース
spid	サーバ・プロセス ID
proxy_suserid	代理のサーバ・ユーザ ID
client_name	set clientname 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 コマンドで設定された、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 コマンドを使用して設定された、クライアントが受け取る日付の形式
is_showplan_on	set showplan がオンの場合は YES、オフの場合は NO
is_noexec_on	no exec がオンの場合は YES、オフの場合は NO

アクセス・ルールと ACF による問題の解決

この項では、ある問題の解決方法を示します。その問題とは、セキュリティ・レベルが異なる 5 人のユーザが、それぞれのユーザのセキュリティ・レベル以下の値を持つローだけを参照できるようにするというものです。この解決方法では、アクセス・ルールを Application Context Facility とともに使用し、Dave というユーザが参照するローだけを表示します。

次の 5 つのログインがあります。

- Anne のセキュリティ・レベルは 1 です。
- Bob のセキュリティ・レベルは 1 です。
- Cassie のセキュリティ・レベルは 2 です。
- Dave のセキュリティ・レベルは 2 です。
- Ellie のセキュリティ・レベルは 4 です。

各ユーザが参照できるローは、rlac の値が自分のセキュリティ・レベル以下であるローだけとなるようにする必要があります。このようにするには、アクセス・ルールを作成して ACF を適用します。

rlac カラムは integer 型、appcontext 引数は char 型です。

```
create access rule rlac_rule as
    @value <= convert(int, get_appcontext("titles",
                "rlac"))

sp_bindrule rlac_rule, "titles.rlac"

/* log in as Dave and apply ACF value of 2*/

select set_appcontext("titles", "rlac", "2")

/*this value persists throughout the session*/
/*select all rows*/

select title_id, rlac from titles
---------------------
```

title_id	rlac
PC8888	1
BU1032	2
PS7777	1
PS3333	1
BU1111	2
PC1035	1
BU2075	2
PS2091	1
PS2106	1
BU7832	2
PS1372	1

```
(11 rows affected)
```

ログイン・トリガの使用

注意　この項の情報の一部は、「Login Triggers in ASE 12.5」(Copyright 1998-2002), Rob Verschoor/ Sypron B.V (http://www.sypron.nl/logtrig.html) からの引用です。

ログイン・トリガは、ユーザがログインするたびに、指定されたストアド・プロシージャを実行します。ログイン・トリガは、バックグラウンドで実行される点を除けば、通常のストアド・プロシージャと同じです。これは、正常なログイン・プロセスの最後のステップとして実行され、ログインするユーザのアプリケーション・コンテキストを設定します。

サーバ内のユーザに対してログイン・トリガを登録できるのは、システム・セキュリティ担当者だけです。

安全な環境を実現するには、システム管理者は次のことを実行する必要があります。

1 set_appcontext 関数に対する select 権限を取り消します。ログイン・トリガの所有者は、sa_role を付与されたユーザであっても、set_appcontext を使用するには明示的なパーミッションが必要です。

2 ストアド・プロシージャからログイン・トリガを設定し、そのログイン・トリガをユーザに登録します。

3 ユーザが実行するログイン・トリガに実行権限を設定します。

ログイン・トリガの作成

ログイン・トリガは、ストアド・プロシージャとして作成します。create trigger コマンドは使用しないでください。次のサンプルでは、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 loginproc as
    declare @appname   varchar(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

grant execute on loginproc to public
go
```

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 sp_modifylogin 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、sp_modifylogin を使用して登録する必要があります。構文は次のとおりです。

```
sp_modifylogin <login_name>, "login script", <sproc_name >
```

- *login_name* − ユーザのログイン名。
- “login script” − このとおりに入力します。“login script” は、次のパラメータ “sproc_name” がログイン・トリガであることを sp_modifylogin に通知します。
- *sproc_name* − このユーザのログイン・トリガとして設定されるストアド・プロシージャの名前。

この手順は、ユーザのデフォルト・データベースで実行します。ログイン・トリガとして登録するストアド・プロシージャは、ユーザのデフォルト・データベース内になければなりません。Adaptive Server はユーザのデフォルト・データベースの sysobjects テーブルでログイン・トリガ・オブジェクトを検索するからです。

ログイン・トリガの設定

次の例では、Adaptive Server ログイン my_login のログイン・トリガとしてストアド・プロシージャ my_proc (設定するデータベース内に存在している必要があります) を設定します。

```
sp_modifylogin my_login, "login script", my_proc
```

この場合も、コマンドはユーザのデフォルト・データベースから実行する必要があります。Adaptive Server では、このストアド・プロシージャに対する execute 権限がログインにあるかどうかの検査が行われますが、ユーザが実際にログインしてログイン・トリガを実行するまでは権限の検査は行われません。

ログイン・トリガの削除と変更

ログイン・トリガとして設定されているストアド・プロシージャを削除することはできません。初めに設定を解除する必要がありますが、それにはログイン・トリガを完全に削除するか、ログイン・トリガの設定を別のストアド・プロシージャに変更します。ログイン・トリガを削除するには、次のように入力します。

```
sp_modifylogin my_login, "login script", NULL
```

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。

```
sp_modifylogin my_login, "login script", diff_proc
```

ログイン・トリガの表示　現在のログイン・トリガを表示するには、sp_displaylogin を使用します。

```
sp_displaylogin my_login
go
(....)
Default Database:my_db
Default Language:
Auto Login Script:my_proc
....
```

ログイン・トリガの実行

ログイン・トリガが通常のストアド・プロシージャと異なるのは、登録されたログイン・トリガは、アクティブなユーザ接続を持たずにバックグラウンドで実行される点です。ログイン・トリガが設定されている場合は、そのユーザがログインすると、Adaptive Server はクライアント・アプリケーションからの何らかのコマンドを実行する前にログイン・トリガをバックグラウンドで自動的に実行します。

1 つのログインで複数の同時接続を確立する場合は、ログイン・トリガはセッションごとに独立して実行されます。同様に、複数のログインが同じストアド・プロシージャをログイン・トリガとして設定することもできます。

ログイン・トリガとして設定されたストアド・プロシージャはバックグラウンドで実行されるので、ストアド・プロシージャの標準機能の中には使用できなくなるものがあります。たとえば、デフォルト値のないパラメータをプロシージャとの間で受け渡すことはできません。また、プロシージャが結果の値を返すことはありません。

この特別な実行モードは、ログイン・トリガのストアド・プロシージャによって呼び出されるすべてのプロシージャと、ログイン・トリガのストアド・プロシージャ自体によって生成されるすべての出力に影響を与えます。

ログイン・トリガのストアド・プロシージャを通常のストアド・プロシージャとして、たとえば、isql から実行することもできます。プロシージャは通常どおりに動作し、出力とエラー・メッセージもすべて通常どおり表示されます。

ログイン・トリガの出力について

ストアド・プロシージャをバックグラウンド・タスクとして実行した場合の最大の影響は、一部のエラー・メッセージと同様に、ログイン・トリガからの出力がクライアント・アプリケーションではなく Adaptive Server エラー・ログ・ファイルに書き込まれることです。

エラー・ログでは、print または raiserror のメッセージの出力は background task message または background task error というテキストで始まります。たとえば、ログイン・トリガ内で print “Hello!” という文と raiserror 123456 という文を実行した場合は、Adaptive Server エラー・ログには次のように出力されます。

```
(....) background task message:Hello!
(....) background task error 123456:This is test message 123456
```

ただし、すべての出力が Adaptive Server エラー・ログに書き込まれるわけではありません。

- select 文の結果セットは、通常であればクライアント接続に送信されますが、この場合は Adaptive Server エラー・ログも含めてどこにも出力されません。この情報は消滅します。
- 正常に実行される文には、insert...select 文と select...into 文の他に、通常は結果セットをクライアント・アプリケーションに送信しないその他の DML 文、および通常のストアド・プロシージャ内で実行可能な DDL 文があります。

その他のアプリケーションでのログイン・トリガの使用

ログイン・トリガは、Adaptive Server のロー・レベル・アクセス制御機能の一部です。したがって、セッションが Adaptive Server にログインした後は、ログイン・トリガをアクセス・ルールおよびアプリケーション・コンテキストと組み合わせて使用することにより、ロー・レベル・アクセス制御を設定することができます。ただし、ログイン・トリガは他の目的で使用することもできます。

同時接続数の制限

次の例では、1 つのログインで確立できる Adaptive Server への同時接続数を制限します。この例の手順 1 と 2 で説明する各コマンドは、アクセス制限の対象となるユーザのデフォルト・データベースで実行されます。

1　システム管理者として、limit_user_sessions ストアド・プロシージャを次のように作成します。

```
create procedure limit_user_sessions
as
declare @cnt int,
   @limit int,
   @loginname varchar(32)

select @limit = 2 -- max nr. of concurrent logins

/* determine current #sessions */
select @cnt = count(*)
from master.dbo.sysprocesses
where suid = suser_id()

/* check the limit */
if @cnt > @limit
begin

    select @loginname = suser_name()
    print "Aborting login [%1!]:exceeds session
        limit [%2!]",
        @loginname, @limit
    /* abort this session */
```

```
    select syb_quit()
end
go

grant exec on limit_user_sessions to public
go
```

2 システム・セキュリティ担当者として、このストアド・プロシージャをユーザ “bob” のログイン・トリガとして設定します。

```
sp_modifylogin "bob", "login script",
"limit_user_sessions"
go
```

3 ユーザ “bob” が Adaptive Server の 3 番目のセッションを作成するとき、syb_quit() 関数を呼び出すログイン・トリガによってこのセッションを終了します。

```
% isql -SASE125 -Ubob -Pbobpassword
1> select 1
2> go

CT-LIBRARY error:
ct_results(): network packet layer: internal net library
error: Net-Library operation terminated due to disconnect
```

4 このメッセージは、Adaptive Server のエラー・ログ・ファイルに記録されます。

```
(...) background task message:Aborting login [
my_login]:exceeds session limit [2]
```

時間ベースの制限の適用

この例では、システム管理者がログイン・トリガを作成して、ユーザ・セッションに対して時間ベースの制限を適用する方法を示します。手順 1 ～ 4 で説明する各コマンドは、アクセス制限の対象となるユーザのデフォルト・データベースで実行されます。

1 システム管理者として次のテーブルを作成します。

```
create table access_times (
suid int not null,
dayofweek tinyint,
shiftstart time,
shiftend time)
```

2 システム管理者として、テーブル access_times に次のようなローを挿入します。これらのローでは、ユーザ “bob” は、月曜日の午前 9 時～午後 5 時に Adaptive Server へのログインを許可され、ユーザ “mark” は、火曜日の午前 9 時～午後 5 時に Adaptive Server へのログインを許可されることが示されています。

```
insert into access_times
select suser_id('bob'), 1, '9:00', '17:00'
go
```

```
insert into access_times
select suser_id('mark'), 2, '9:00', '17:00'
go
```

3　システム管理者として limit_access_time ストアド・プロシージャを作成します。このストアド・プロシージャでは、access_time テーブルを参照して、ログイン・アクセスを許可するかどうかを決定します。

```
create procedure limit_access_time as
declare @curdate date,
    @curdow tinyint,
    @curtime time,
    @cnt int,
    @loginname varchar(32)

-- setup variables for current day-of-week, time
select @curdate = current_date()
select @curdow = datepart(cdw,@curdate)
select @curtime = current_time()
select @cnt = 0

-- determine if current user is allowed access
select @cnt = count(*)
from access_times
where suid = suser_id()
and dayofweek = @curdow
and @curtime between shiftstart and shiftend

if @cnt = 0
begin
   select @loginname = suser_name()
   print "Aborting login [%1!]:login attempt past
     normal working hours", @loginname

   -- abort this session
   return -4
end
go

grant exec on limit_access_time to public
go
```

4　システム・セキュリティ担当者として、limit_access_time ストアド・プロシージャをユーザ “bob” とユーザ “mark” のログイン・トリガとして設定します。

```
sp_modifylogin "bob", "login script",
"limit_access_time"
go
sp_modifylogin "mark", "login script",
"limit_access_time"
go
```

5 月曜日に、ユーザ “bob” はセッションを正常に作成できます。

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

しかし、ユーザ “mark” の Adaptive Server へのアクセスは拒否されます。

```
isql -Umark -Ppassword
1> select 1
2> go
CT-LIBRARY error:
ct_results(): network packet layer: internal net library
error: Net-Library operation terminated
due to disconnect
```

6 次のメッセージがエラー・ログに書き込まれます。

```
(...) server back-ground task message:Aborting login
[mark]:login attempt past normal working hours
```

上記の例では、特定のログインの同時接続数を制限し、このログインのアクセスを特定の時間帯だけに制限しました。ただし、欠点が 1 つあります。それは、セッションが終了した理由をクライアント・アプリケーションが容易に検出できないことです。ユーザに、たとえば「ユーザ数が多すぎます。後でやり直してください」などのメッセージを表示するには、別の方法を使用します。

現在のセッションを終了させるだけの組み込み関数 syb_quit() を呼び出す代わりに、ストアド・プロシージャ内でエラーを発生させて、ログイン・トリガのストアド・プロシージャをアボートします。

たとえば、ゼロ除算を行うとログイン・トリガのストアド・プロシージャがアボートし、セッションが終了して、メッセージが表示されます。

ログイン・トリガの制限事項

次のアクションは制限を受けます。

- ログイン・トリガを使用して set nocount on や set rowcount on などのセッション固有のオプションを設定することはできません。ストアド・プロシージャ内で設定したセッション・オプションが有効であるのは、そのストアド・プロシージャ内のみです。
- #temp テーブルを作成して後でそのセッション内で使用することはできません。他のストアド・プロシージャの場合と同様に、プロシージャが完了すると #temp テーブルは自動的に削除され、元のセッション設定がリストアされます。

- sa ログインにはログイン・トリガを使用しないでください。ログイン・トリガが失敗すると、Adaptive Server からロック・アウトされる場合があります。
- 数秒以上かかるような処理をログイン・トリガで実行すると処理の問題が生じる場合があるので、そのような処理にはログイン・トリガを使用しないでください。

問題と情報

- Adaptive Server エラー・ログにアクセスできない場合は、ログイン・トリガを使用しないでください。常に Adaptive Server エラー・ログでエラー・メッセージを確認してください。
- Adaptive Server バージョン 15.0.2 以降では、ログイン・トリガでエクスポート可能なオプションを設定または解除すると、サーバが起動する時点のログイン・プロセスで反映されます。

  この動作を無効にするには、ログイン・トリガ内で set export_options off を実行します。

  Adaptive Server バージョン 15.0.1、12.5.4、およびそれ以前では、ログイン・トリガのオプションを有効にするには、トレース・フラグ 4073 を有効にして Adaptive Server を起動する必要があります。
- isql などのクライアント・アプリケーションは、ログイン・トリガの存在や実行を認識しません。ログインに成功すると、すぐにクライアント・アプリケーションのコマンド・プロンプトが表示されますが、Adaptive Server によってコマンドが実行されるのはログイン・トリガが正常に実行された後です。この isql のプロンプトは、ログイン・トリガによってユーザ接続が終了した場合でも表示されます。
- Adaptive Server にログインするユーザには、ログイン・トリガのストアド・プロシージャを使用するための execute パーミッションが必要です。execute パーミッションが付与されていない場合は、Adaptive Server のエラー・ログにエラー・メッセージが出力され、ユーザ接続はただちに終了します (ただし、isql のコマンド・プロンプトは表示されます)。

  Adaptive Server のエラー・ログには、次のようなメッセージが出力されます。

  ```
  EXECUTE permission denied on object my_proc, database
  my_db, owner dbo
  ```
- ログイン・トリガのストアド・プロシージャのパラメータには、必ずデフォルト値を設定してください。ストアド・プロシージャのパラメータの中にデフォルト値がないものが見つかると、ログイン・トリガは失敗し、Adaptive Server のエラー・ログに次のようなエラーが出力されます。

  ```
  Procedure my_proc expects parameter @param1, which was not
  supplied...
  ```

ログイン・トリガに対する実行権限の無効化

データベース所有者または管理者は、ログイン・トリガに対する execute 権限を無効化することができます。あるいは、特定の場合にのみアクセスを許可するようにログイン・トリガをコーディングすることもできます。たとえば、データベース所有者または管理者がテーブルを更新している間は、一般のユーザがサーバを使用できないようにする場合です。

注意 ログイン・トリガが負の数を返した場合は、ログインは失敗です。

ログイン・トリガからの set オプションのエクスポート

Adaptive Server では、ログイン・トリガ内の set コマンドのオプションをユーザ・セッション全体で有効にできます。

次の set オプションは自動的にエクスポートされます。

- showplan
- arithabort [overflow | numeric_truncation]
- arithignore [overflow]
- colnames
- format
- statistics io
- procid
- rowcount
- altnames
- nocount
- quoted_identifier
- forceplan
- fmtonly
- close on endtran
- fipsflagger
- self_recursion
- ansinull
- dup_in_subquery
- or_strategy

- flushmessage
- ansi_permissions
- string_rtruncation
- prefetch
- triggers
- replication
- sort_resources
- transactional_rpc
- cis_rpc_handling
- strict_dtm_enforcement
- raw_object_serialization
- textptr_parameters
- remote_indexes
- explicit_transaction_required
- statement_cache
- command_status_reporting
- proc_return_status
- proc_output_params

グローバル・ログイン・トリガの設定

グローバル・ログイン・トリガを設定するには、sp_logintrigger を使用します。これは、ユーザのログインごとに実行されます。ユーザ固有のアクションを取得するには、sp_modifylogin または sp_addlogin を使用してユーザ固有のログイン・トリガを設定します。

注意　トレース・フラグ -T4073 を設定して、このオプションをアクティブ化できます。

第 18 章

監査

この章では、インストール環境に応じた監査の設定方法について説明します。

Adaptive Server での監査の概要

安全なシステムを構築するうえで重要な要素は、責任の所在を明確にすることです。この責任を確実に保つ手段の 1 つとして、システムのイベントを監査する方法があります。Adaptive Server で発生する多くのイベントは記録が可能です。

監査は、データベース管理システムのセキュリティの重要な機能です。監査証跡を使用して、システムへの侵入とリソースの不正使用を検出します。システム・セキュリティ担当者は、監査証跡を調べることによって、データベース内のオブジェクトに対するアクセスのパターンを調べて特定のユーザのアクティビティをモニタできます。監査レコードを追跡すればユーザを特定できるので、システムを不正に使用しようとするユーザに対する抑止力となります。

各監査レコードには、イベントの性質、日時、イベントの責任者、イベントが正常か失敗かについて記録できます。監査できるイベントには、ログインとログアウト、サーバの起動、データ・アクセス・コマンドの使用、特定オブジェクトへのアクセス、特定ユーザのアクションなどがあります。「監査証跡」(監査レコードのログ) によって、システム・セキュリティ担当者はシステムで発生したイベントを再構築し、イベントの影響を判断できます。

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

- サーバ全体にわたるセキュリティ関連イベント
- データベース・オブジェクトの作成、削除、変更

- 特定ユーザが行ったすべてのアクション、または特定の役割をアクティブにしてユーザが行ったすべてのアクション
- データベース・アクセス権の付与または取り消し
- データのインポートまたはエクスポート
- ログインとログアウト

Adaptive Server とオペレーティング・システムの監査レコードの関連付け

Adaptive Server の監査レコードをオペレーティング・システムの監査レコードにリンクするには、Adaptive Server のログイン名をオペレーティング・システムのログイン名と同じにするのが最も簡単です。

あるいは、システム・セキュリティ担当者が、ユーザのオペレーティング・システム・ログイン名をそのユーザの Adaptive Server ログイン名にマッピングすることもできます。ただし、この方法では、新規ユーザのログイン名を手作業で登録しなければならず、運用中の保守が必要となります。

監査システム

監査システムは、次のものからなります。

- グローバル監査オプションと監査証跡を含む sybsecurity データベース
- 監査証跡に書き込まれる前の監査レコードが格納される、メモリ内の監査キュー
- 監査を管理するための設定パラメータ
- 監査を管理するためのシステム・プロシージャ

sybsecurity データベース

sybsecurity データベースは、監査機能のインストール・プロセス中に作成されます。model データベース内のすべてのシステム・テーブルの他に、このデータベースには、サーバ全体の監査オプション追跡用のシステム・テーブル sysauditoptions と監査証跡用のシステム・テーブルが含まれます。

sysauditoptions の内容は、グローバル監査オプションの現在の設定値です。これは、ディスク・コマンド、リモート・プロシージャ・コール、独自のユーザ定義監査レコード、またはすべてのセキュリティ関連イベントに対する監査を有効にするかどうかなどを設定するものです。これらのオプションは Adaptive Server 全体に影響します。

監査証跡

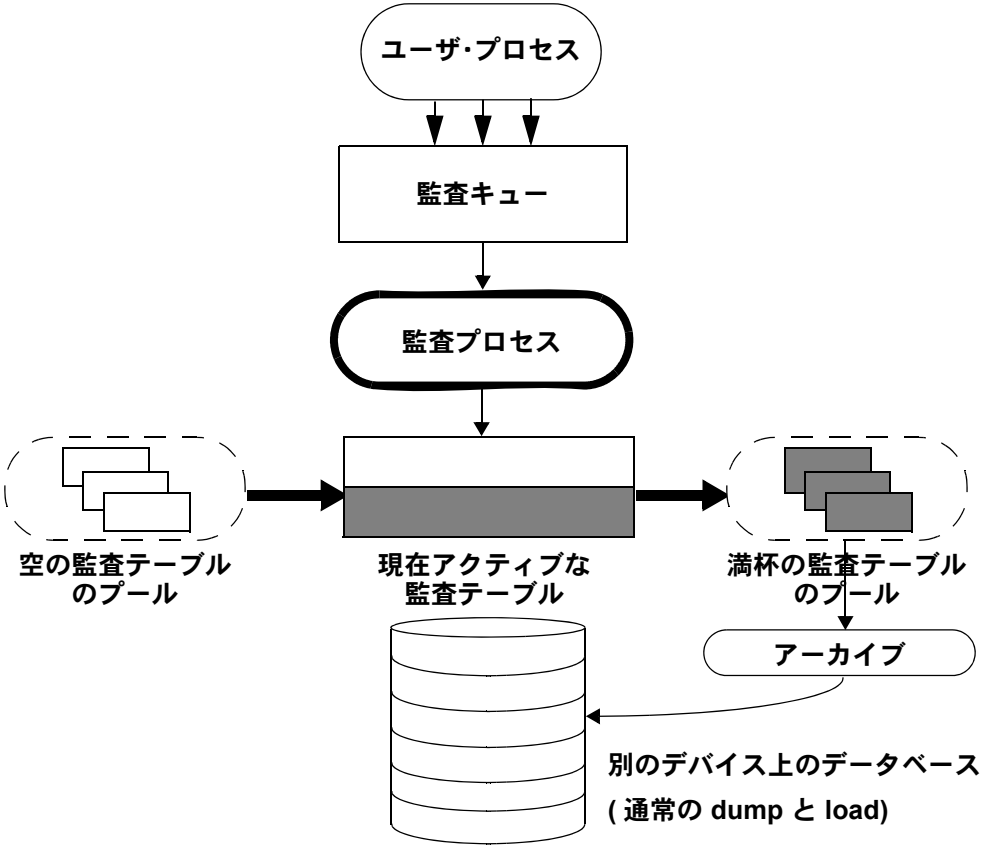
Adaptive Server は、sysaudits_01 から sysaudits_08 までのシステム・テーブルに監査証跡を格納します。監査機能をインストールするときに、インストール環境に合わせた監査テーブルの数を決定します。たとえば、2 つの監査テーブルを使用する場合は、その名前は sysaudits_01 と sysaudits_02 となります。「現在の監査テーブル」は、常に 1 つしかありません。Adaptive Server は、現在の監査テーブルにすべての監査データを書き込みます。システム・セキュリティ担当者は sp_configure を使用して、どの監査テーブルを現在のものにするかを設定したり、変更したりできます。

監査テーブル数は 2 以上とし、各テーブルを個別の監査デバイス上に置くことをおすすめします。このようにすれば、監査レコードが失われることや手動介入を必要とすることなく、監査テーブルがアーカイブされ、処理されるので、監査プロセスはスムーズに実行されるようになります。

警告！ Sybase では、運用システムに対して単一の監査テーブルを使用しないよう強くおすすめします。使用する監査テーブルが 1 つだけの場合は、監査レコードが失われるおそれがあります。システム・リソースの制約から単一の監査テーブルしか使用できない場合は、「単一テーブル監査」(624 ページ) の指示を参照してください。

図 18-1 は、監査プロセスが複数の監査テーブルをどのように処理するかを示しています。

図 *18-1:* 複数の監査テーブルを使用した監査



監査システムは、メモリ内監査キューから現在の監査テーブルに監査レコードを書き込みます。現在の監査テーブルが満杯に近づいたときに、スレッショルド･プロシージャによってそのテーブルを自動的に別のデータベースにアーカイブできます。アーカイブ・データベースは、dump コマンドと load コマンドによってバックアップおよびリストアできます。バックアップからアーカイブされた監査テーブルに対して読み取り専用アクセスを行うには、アーカイブ・データベースへのアクセスを使用します。『システム管理ガイド 第 2 巻』の「第 14 章 アーカイブ・データベースへのアクセス」を参照してください。監査証跡の管理の詳細については、「監査証跡の管理の設定」(616 ページ) を参照してください。

監査キュー

監査イベントが発生すると、監査レコードは、まずメモリ内の監査キューに格納されます。このレコードは、監査プロセスによって監査証跡に書き込まれるまで、メモリ内に残ります。監査キューのサイズは、sp_configure の audit queue size パラメータを使用して設定できます。

監査キューのサイズを設定するにあたっては、システム・クラッシュ時にキュー内のレコードが失われる危険性と、キューが満杯になったときのパフォーマンスのロスとのトレードオフについて考慮してください。監査レコードがキュー内にあるかぎり、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 査レコードは監査証跡に直接書き込まれるのではないので、監査レコードが現在の監査テーブルにすぐに保管されるとは考えないでください。

監査設定パラメータ

監査プロセスの管理には、次の設定パラメータを使用します。

- auditing は、Adaptive Server 全体の監査を有効または無効にします。このパラメータは、sp_configure の実行後すぐに反映されます。このパラメータが有効な場合にのみ監査が実行されます。
- audit queue size は、監査キューのサイズを設定します。このパラメータは、メモリの割り付けに影響を与えるため、Adaptive Server が再起動されるまでは有効になりません。
- suspend audit when device full は、監査デバイスが満杯になったときの監査プロセスの動作を制御します。このパラメータは、sp_configure の実行後すぐに反映されます。
- current audit table は、現在の監査テーブルを設定します。このパラメータは、sp_configure の実行後すぐに反映されます。

監査用のシステム・プロシージャ

監査プロセスの管理には、次のシステム・プロシージャを使用します。

- sp_audit は、監査オプションを有効または無効にします。監査対象のイベントを指定するのに必要なシステム・プロシージャはこれだけです。
- sp_displayaudit は、アクティブな監査オプションを表示します。
- sp_addauditrecord は、監査証跡にユーザ定義監査レコード (コメント) を追加します。ユーザがこの種のレコードを追加できるのは、システム・セキュリティ担当者が sp_audit を使って独自の監査を有効にした場合のみです。

監査のインストールと設定

表 18-1 は、監査を設定するための一般的な手順を示しています。

表 *18-1:* 監査を行うための一般的な手順

アクション	説明	参照箇所
1. 監査をインストールする。	監査テーブル数を設定する。監査証跡および sybsecurity データベース内の syslogs トランザクション・ログにデバイスを割り当てる。	「監査システムのインストール」(612 ページ) と、Adaptive Server の『インストール・ガイド』および『設定ガイド』を参照。
2. 監査証跡の管理を設定する。	現在の監査テーブルがほとんど満杯になったときに制御を受け取るスレッショルド・プロシージャを作成して設定する。このプロシージャは、自動的に新しい監査テーブルに切り替えて、現在のテーブルの内容をアーカイブする。 また、この手順では、audit queue size と suspend audit when device full の各設定パラメータも設定する。	「監査証跡の管理の設定」(616 ページ) 単一テーブルでの監査については、「単一テーブル監査」(624 ページ) を参照。
3. sybsecurity データベース内のトランザクション・ログの管理を設定する。	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する。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る。	「トランザクション・ログの管理の準備」(622 ページ)
4. 監査オプションを設定する。	sp_audit を使用して、監査対象のイベントを設定する。	「グローバル監査オプションの設定」(628 ページ)
5. 監査を有効にする。	sp_configure を使用して auditing 設定パラメータをオンにする。Adaptive Server は、現在の監査テーブルへの監査レコードの書き込みを開始する。	「監査の有効化と無効化」(623 ページ)
6. 監査を再起動する。	監査が失敗した場合は、sp_audit restart を使用して監査を再起動する。	「監査の再起動」(627 ページ)

監査システムのインストール

監査システムは、通常、Sybase インストール・プログラムの auditinit を使用してインストールします。あるいは、auditinit を使用しないで監査システムをインストールすることもできます。詳細については、「installsecurity による監査のインストール」(613 ページ) を参照してください。インストールと auditinit については、プラットフォームの『Adaptive Server インストール・ガイド』および『Adaptive Server 設定ガイド』を参照してください。

監査機能をインストールするときに、監査証跡に使用するシステム・テーブルの数、各監査システム・テーブル用のデバイス、syslogs トランザクション・ログ用のデバイスを設定できます。

監査証跡のためのテーブルとデバイス

指定できるシステム・テーブルは最高 8 つです (sysaudits_01 から sysaudits_08 まで)。監査証跡には、少なくとも 2 つのテーブルを使用するようにしてください。各テーブルは、マスタ・デバイスとは別に独自のデバイスに配置します。このようにすると、スレッショルド・プロシージャを使用して、現在の監査テーブルが満杯になる前にその内容を自動的にアーカイブしてから、新しい空のテーブルに切り替えてそれ以降の監査レコードを保存することができます。

syslogs トランザクション・ログ・テーブルのデバイス

監査機能をインストールするときに、syslogs システム・テーブルで構成されるトランザクション・ログ用に個別のデバイスを指定する必要があります。この syslogs テーブルは、すべてのデータベースに存在するもので、そのデータベースで実行されるトランザクションのログが格納されます。

installsecurity による監査のインストール

$SYBASE/ASE-15_0/scripts ディレクトリに、監査機能をインストールするためのスクリプトである *installsecurity* があります。

注意　この例では、サーバが使用する論理ページ・サイズは 2K であるとします。

installsecurity を使用して監査機能をインストールするには、次の手順に従います。

1　disk init コマンドと create database コマンドを使用して、監査デバイスと監査データベースを作成します。例：

```
disk init name = "auditdev",
    physname = "/dev/dsk/c2d0s4",
    size = "10M"
disk init name = "auditlogdev",
    physname = "/dev/dsk/c2d0s5",
    size = "2M"
create database sybsecurity on auditdev
    log on auditlogdev
```

2　isql を使用して、*installsecurity* スクリプトを実行します。

```
cd $SYBASE/ASE-12_5/scripts
setenv DSQUERY server_name
isql -Usa -Ppassword -Sserver_name < installsecurity
```

3　Adaptive Server を停止して再起動します。

これらの手順を終了すると、sybsecurity データベースの独自セグメントに 1 つの監査テーブル (sysaudits_01) が作成されます。この時点で監査を有効にすることは可能ですが、sp_addaudittable システム・プロシージャを使用して、さらに監査テーブルを追加する必要があります。disk init、create database、sp_addaudittable の詳細については、『リファレンス・マニュアル：プロシージャ』を参照してください。

複数デバイスへの監査データベースの移動

sybsecurity データベースは、master データベースとは別の独自のデバイス上に置く必要があります。複数の監査テーブルがある場合は、テーブルをそれぞれ専用のデバイスに配置します。各テーブルを別のセグメントに置き、それぞれが別のデバイスを指すようにすると便利です。現在 master と同じデバイスに sybsecurity がある場合、または何らかの理由で別のデバイスに sybsecurity を移動したい場合は、以下の項で説明する手順のいずれかを使用してください。データベースを移動する場合、既存のグローバル監査設定を保存するかどうかを指定できます。

グローバル監査設定を保存しない *sybsecurity* の移動

注意 この手順には sybsecurity データベースの削除が含まれます。この削除によって、sybsecurity で記録されている監査レコードとグローバル監査設定がすべて破棄されます。sybsecurity データベースを削除する前に、必ず、バックアップによって、または「監査テーブルのアーカイブ」(617 ページ) の手順に従って既存のレコードをアーカイブし、sybsecurity テーブルに残っている履歴データを保管してください。

グローバル監査設定を保存しないで、sybsecurity データベースを移動するには、次の手順に従います。

1 次のコマンドを実行し、ログインに関連する情報を syslogins システム・テーブルから削除します。

```
sp_audit "all","all","all","off"
```

2 sybsecurity データベースを削除します。

3 次で説明しているいずれかのインストール手順に従い、sybsecurity をもう一度インストールします。

- 使用しているプラットフォームの設定ガイド
- 「installsecurity による監査のインストール」(613 ページ)

4 このインストール・プロセスで、sybsecurity データベースを、必ずマスタ・デバイスとは別の 1 つまたは複数のデバイス上に置くようにしてください。

sybsecurity の移動とグローバル監査設定の保存

❖ *sybsecurity* データベースを移動して、グローバル監査設定を保存するには、次の手順に従います。

1 sybsecurity データベースをダンプします。

```
dump database sybsecurity to "/remote/sec_file"
```

2 sybsecurity データベースを削除します。

```
drop database sybsecurity
```

3 sybsecurity データベースを配置する最初のデバイスを初期化します。

```
disk init name = "auditdev",
    physname = "/dev/dsk/c2d0s4",
    size = "10M"
```

4 セキュリティ・ログを配置するデバイスを初期化します。

```
disk init name = "auditlogdev",
    physname = "/dev/dsk/c2d0s5",
    size = "2M"
```

5 新しい sybsecurity データベースを作成します。

```
create database sybsecurity on auditdev
    log on auditlogdev
```

6 古い sybsecurity データベースの内容を、新しく作成したデータベースにロードします。グローバル監査設定は維持されます。

```
load database sybsecurity from "/remote/sec_file"
```

7 online database コマンドを実行します。このコマンドは、必要に応じて sysaudits と sysauditoptions をアップグレードします。

```
online database sybsecurity
```

8 プラットフォームの『Adaptive Server Enterprise 設定ガイド』に従って、監査システム・プロシージャをロードします。

❖ 複数の *sysaudits* テーブルを *sybsecurity* に作成するには、次の手順に従います。

1 追加テーブルを配置するデバイスを初期化します。

```
disk init name = "auditdev2",
    physname = "/dev/dsk/c2d0s6",
    size = "10M"
```

2 手順 1 で初期化したデバイスに sybsecurity データベースを拡張します。

```
alter database sybsecurity on auditdev2 = "2M"
```

3 sp_addaudittable システム・プロシージャを実行して、手順 1 で初期化したデバイス上に次の sysaudits テーブルを作成します。

```
sp_addaudittable auditdev2
```

4 各 sysaudits テーブルに対して、1 ～ 3 の手順を繰り返します。

監査証跡の管理の設定

監査証跡を効率的に管理するには、次の手順に従います。

1 監査機能が、個別のデバイスに配置された複数のテーブルを使用するようにインストールされていることを確認します。そうでない場合は、監査テーブルとデバイスを追加する必要があります。

2 スレッショルド・プロシージャを作成して、各監査テーブル・セグメントに付加します。

3 監査キュー・サイズと、現在の監査テーブルが満杯になった場合の適切な操作を示す設定パラメータを設定します。

以下の各項では、個別のデバイスに配置された複数テーブルを使用するように監査機能をインストールしたものと想定しています。監査テーブル用のデバイスが 1 つしかない場合は、「単一テーブル監査」(624 ページ) へ進んでください。

スレッショルド・プロシージャの設定

監査を有効にする前に、スレッショルド・プロシージャを設定して、現在のテーブルが満杯になったら監査テーブルを自動的に切り替えるようにしてください。

監査デバイス・セグメントのスレッショルド・プロシージャは、次のタスクを実行する必要があります。

- sp_configure を使用して current audit table 設定パラメータを設定し、次の空白の監査テーブルを現在のテーブルにする。
- insert...select コマンドを使用して、満杯に近づいた監査テーブルをアーカイブする。

現在の監査テーブルの変更

current audit table 設定パラメータは、Adaptive Server が監査ローを書き込むテーブルを設定します。システム・セキュリティ担当者は、sp_configure を実行して現在の監査テーブルを変更できます。構文は次のとおりです。*n* は、新しい現在の監査テーブルを指定する整数です。

```
sp_configure "current audit table", n
  [, "with truncate"]
```

n の有効な値は次のとおりです。

- 1 は sysaudits_01、2 は sysaudits_02 を示します。
- 0 は、次のテーブルを自動的に現在の監査テーブルとして設定するように Adaptive Server に指示します。たとえば、インストール環境に 3 つの監査テーブル sysaudits_01、sysaudits_02、sysaudits_03 がある場合、現在の監査テーブルは次のように設定されます。
  - 現在の監査テーブルが sysaudits_01 の場合は 2
  - 現在の監査テーブルが sysaudits_02 の場合は 3
  - 現在の監査テーブルが sysaudits_03 の場合は 1

with truncate オプションは、新しいテーブルが空でない場合に、そのテーブルをトランケートすることを指定します。このオプションを指定しないと、テーブルが空になっていない場合、sp_configure コマンドは失敗します。

注意　Adaptive Server が現在の監査テーブルをトランケートしたときに、データがアーカイブ済みでなければ、そのテーブルの監査レコードは失われます。with truncate オプションを使用する前に、必ず監査データをアーカイブするようにしてください。

---

sp_configure を実行して現在の監査テーブルを変更するには、sso_role をアクティブにしてください。スレッショルド・プロシージャを作成して、現在の監査テーブルを自動的に変更することもできます。

監査テーブルのアーカイブ

select とともに insert を使用すると、sybsecurity 内の監査テーブルと同じカラムを持つ既存のテーブルに、監査データをコピーすることができます。

スレッショルド・プロシージャが、別のデータベース内のアーカイブ・テーブルにデータを正常にコピーできるようにするには、次の準備手順を実行してください。

1　sybsecurity 内の監査テーブルが存在するデバイスとは別のデバイス上に、アーカイブ・データベースを作成します。

2　sybsecurity の監査テーブルと同じカラムを持つアーカイブ・テーブルを作成します。このようなテーブルが存在しない場合は、select into を使用して where 句に false の条件を指定することによって、空のテーブルを作成することができます。例：

```
use aud_db
go
select *
    into audit_data
    from sybsecurity.dbo.sysaudits_01
    where 1 = 2
```

where 条件は常に false です。したがって、sysaudits_01 の複製である空のテーブルが作成されます。

select into を使用する前に、アーカイブ・データベースで sp_dboption を使用して select into/bulk copy データベース・オプションをオンにしておく必要があります。

スレッショルド・プロシージャでは、sp_configure を使用して監査テーブルを変更した後で、insert と select を使用して、アーカイブ・データベース内のアーカイブ・テーブルにデータをコピーします。このプロシージャで実行するコマンドの例を示します。

```
insert aud_db.sso_user.audit_data
select * from sybsecurity.dbo.sysaudits_01
```

監査セグメント用スレッショルド・プロシージャの例

次のスレッショルド・プロシージャの例では、監査用に 3 つのテーブルが設定されているものと想定しています。

```
declare @audit_table_number int
/*
** Select the value of the current audit table
*/
select @audit_table_number = scc.value
from master.dbo.syscurconfigs scc, master.dbo.sysconfigures sc
where sc.config=scc.config and sc.name  = “current audit table”
/*
** Set the next audit table to be current.
** When the next audit table is specified as 0,
** the value is automatically set to the next one.
*/
exec sp_configure “current audit table”, 0, “with truncate”
/*
** Copy the audit records from the audit table
** that became full into another table.
*/
if @audit_table_number = 1
    begin
        insert aud_db.sso_user.sysaudits
            select * from sysaudits_01
        truncate table sysaudits_01
    end
else if @audit_table_number = 2
    begin
        insert aud_db.sso_user.sysaudits
            select * from sysaudits_02
        truncate table sysaudits_02
    end
return(0)
```

各監査セグメントへのスレッショルド・プロシージャの付加

スレッショルド・プロシージャを各監査テーブル・セグメントに付加するには、システム・プロシージャ sp_addthreshold を使用します。

sp_addthreshold を実行する前に、必ず次のことを行ってください。

- インストール環境に合わせて設定する監査テーブルの数と、そのデバイス・セグメントの名前を決定する。
- sp_addthreshold の実行に必要な、スレッショルド・プロシージャに含まれるすべてのコマンドに対するパーミッションおよび役割を用意する。

警告！ sp_addthreshold と sp_modifythreshold は、sa_role を直接付与されたユーザだけがスレッショルドを追加または変更できるようにするために検査を行います。スレッショルドを追加または変更するときにアクティブなすべてのシステム定義の役割が、そのログインに有効な役割として、systhresholds テーブルに挿入されます。ただし、スレッショルド・プロシージャの起動時には、直接付与された役割だけがアクティブになります。

監査テーブルとそのセグメント

監査機能をインストールするとき、auditinit によって各監査テーブルの名前とそのセグメントが表示されます。セグメント名は、sysaudits_01 では “aud_seg1”、sysaudits_02 では “aud_seg2” というようになります。sybsecurity を現在のデータベースとして sp_helpsegment を実行すると、sybsecurity データベース内のセグメントに関する情報を検索できます。インストール環境の監査テーブル数を検索する方法の 1 つとして、次の SQL コマンドがあります。

```
use sybsecurity
go
select count(*) from sysobjects
    where name like "sysaudit%"
go
```

次の SQL コマンドを実行して、監査テーブルと sybsecurity データベースに関する詳細情報を取得することもできます。

```
sp_helpdb sybsecurity
go
use sybsecurity
go
sp_help sysaudits_01
go
sp_help sysaudits_02
go
  ...
```

必要な役割とパーミッション

sp_addthreshold は、データベース所有者かシステム管理者でなければ実行できません。通常は、システム・セキュリティ担当者が、sybsecurity データベースの所有者です。したがって、システム・セキュリティ担当者は sp_addthreshold を実行できます。また、sp_addthreshold を実行する権限に加えて、スレッショルド・プロシージャ内のすべてのコマンドの実行パーミッションが必要です。たとえば、sp_configure を実行して current audit table を設定するには、sso_role がアクティブでなければなりません。スレッショルド・プロシージャが起動すると、Adaptive Server は、sp_addthreshold の実行時に有効であったすべての役割とパーミッションをオンにしようとします。

3 つのデバイス・セグメントにスレッショルド・プロシージャ audit_thresh を付加する方法は次のとおりです。

```
use sybsecurity
go
sp_addthreshold sybsecurity, aud_seg_01, 250, audit_thresh
sp_addthreshold sybsecurity, aud_seg_02, 250, audit_thresh
sp_addthreshold sybsecurity, aud_seg_03, 250, audit_thresh
go
```

このサンプル・スレッショルド・プロシージャ audit_thresh は、現在の監査テーブル内に残っている空きページが 250 よりも少なくなると、制御を受け取ります。

スレッショルド・プロシージャの追加の詳細については、『システム管理ガイド 第 2 巻』の「第 16 章 スレッショルドによる空き領域の管理」を参照してください。

サンプル・スレッショルド・プロシージャによる監査

監査が有効化されると、Adaptive Server は、すべての監査データを最初の現在の監査テーブルである sysaudits_01 に書き込みます。sysaudits_01 の空きページが 250 ページ以内になると、スレッショルド・プロシージャ audit_thresh が起動します。このプロシージャが現在の監査テーブルを sysaudits_02 に切り替えると、その直後から新しい監査レコードは sysaudits_02 に書き込まれます。また、このプロシージャは、sysaudits_01 のすべての監査データを audit_db にあるアーカイブ・テーブル audit_data にコピーします。監査テーブルの巡回は、このように手動介入なしで続きます。

設定パラメータの設定

監査機能をインストールした場合は、次の設定パラメータを設定してください。

- audit queue size は、メモリ内の監査キューのレコード数を設定します。
- suspend audit when device full は、現在の監査テーブルの空きがまったくなくなったときの Adaptive Server の動作を決定します。満杯状態は、現在のテーブル・セグメントに付加されたスレッショルド・プロシージャが正しく機能していない場合にのみ起こります。

監査キューの設定

監査キューのデフォルト・サイズは 100 バイトです。監査キュー・プールが使用するメモリ量は、audit queue size パラメータで定義され、メモリ・プールのデータ・バッファとオーバーヘッドが含まれます。ただし、プールのメモリ量はリリースとチップ・アーキテクチャ間で異なる場合があります。

監査キューの長さを設定するには、sp_configure を使用します。構文は次のとおりです。

sp_configure "audit queue size", [*value*]

value は、監査キューが保持できるレコードの数です。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
```

監査キューのサイズやその他の設定パラメータの設定の詳細は、「第 5 章 設定パラメータ」を参照してください。

デバイスが満杯の場合の監査の中断

複数の監査テーブルがそれぞれマスタ・デバイス以外の独立したデバイス上にあり、各監査テーブル・セグメントにスレッショルド・プロシージャが付加されていれば、監査デバイスが満杯になる状態は決して発生しません。スレッショルド・プロシージャが正常に機能していない場合だけ、「満杯」状態が発生します。デバイスが満杯になったときの処置を指定するには、sp_configure を使用して suspend audit when device full パラメータを設定します。次のいずれかのオプションを選択してください。

- 監査プロセスと、監査可能イベントを生成するすべてのユーザ・プロセスを中断します。システム・セキュリティ担当者が現在の監査テーブルをクリアしてから、通常の操作を再開します。
- 次の監査テーブルをトランケートし、そのテーブルの使用を開始します。これによって、システム・セキュリティ担当者の介入なしに通常の操作を進めることができます。

sp_configure を使用して、この設定パラメータを設定します。また、sso_role をアクティブにする必要があります。構文は次のとおりです。

```
sp_configure "suspend audit when device full",
    [0|1]
```

- 0 を指定すると、現在の監査テーブルが満杯になったときは、次の監査テーブルがトランケートされ、そのテーブルが現在の監査テーブルとして使用されます。このパラメータを 0 に設定しても監査プロセスが中断することはありません。ただし、古い監査レコードは、アーカイブされていなければ完全に消失します。

- 1 (デフォルト値) を指定すると、監査プロセスと監査可能なイベントを生成するすべてのユーザ・プロセスが中断します。通常の操作を再開するには、システム・セキュリティ担当者がログインして、空のテーブルを現在の監査テーブルとして設定する必要があります。この間、システム・セキュリティ担当者は、通常の監査の対象外となります。通常の操作であれば監査レコードが生成されるようなアクションをシステム・セキュリティ担当者が実行すると、そのイベントに関するエラー・メッセージと情報が Adaptive Server のエラー・ログに送信されます。

スレッショルド・プロシージャが監査テーブル・セグメントに付加されている場合は、suspend audit when device full を 1 (on) に設定します。このパラメータを 0 (off) に設定すると、スレッショルド・プロシージャによって監査レコードがアーカイブされる前に、満杯の監査テーブルがトランケートされることがあります。

トランザクション・ログの管理の準備

この項では、sybsecurity 内のトランザクション・ログを管理するためのガイドラインを説明します。

trunc log on chkpt データベース・オプションがアクティブの場合は、自動 checkpoint の実行のたびに syslogs がトランケートされます。監査がインストールされると trunc log on chkpt の値は on になりますが、sp_dboption を使用すると、この値を変更できます。

トランザクション・ログのトランケーション

sybsecurity データベースに対して trunc log on chkpt オプションを有効にすれば、トランザクション・ログが満杯になることはありません。Adaptive Server がチェックポイントを実行するたびに、ログがトランケートされます。このオプションが有効の場合、dump transaction を使用してトランザクション・ログをダンプすることはできませんが、dump database を使用してデータベースをダンプできます。

「スレッショルド・プロシージャの設定」(616 ページ) の手順に従った場合は、監査テーブルは別のデータベース内のテーブルに自動的にアーカイブされます。このアーカイブ・データベースには、標準のバックアップとリカバリの手順を使用できます。

sybsecurity デバイスがクラッシュした場合は、データベースを再ロードして、監査を再開します。最悪の場合でも、メモリ内の監査キューと現在の監査テーブルが失われるだけで済みます。これは、アーカイブ・データベースにそれ以外の監査データがすべて含まれるためです。データベースを再ロードしたら、sp_configure with truncate を使用して、現在の監査テーブルを設定してトランケートします。

データベースをダンプした後に、サーバ全体の監査オプションを変更していなければ、sysauditoptions に保管されているすべての監査オプションが、sybsecurity の再ロード時に自動的にリストアされます。変更した場合は、監査を再開する前にスクリプトを実行してオプションを設定します。

トランケーションを使用しないトランザクション・ログの管理

db_option を使用して trunc log on chkpt をオフにすると、トランザクション・ログが満杯になる可能性があります。「ラストチャンス・スレッショルド・プロシージャ」をトランザクション・ログ・セグメントに付加することを検討してください。このプロシージャは、セグメントの空き領域が、Adaptive Server によって自動的に計算されるスレッショルドの量を下回ると起動されます。スレッショルド量は、トランザクション・ログのバックアップに必要な空きログ・ページ数から計算されます。

ラストチャンス・スレッショルド・プロシージャのデフォルト名は sp_thresholdaction ですが、sa_role がアクティブになっていれば sp_modifythreshold を使用して別の名前を指定できます。

注意　sp_modifythreshold は、“sa_role” がアクティブであることをチェックします。詳細については、「各監査セグメントへのスレッショルド・プロシージャの付加」(619 ページ) を参照してください。

Adaptive Server のデフォルトのプロシージャはありませんが、『システム管理ガイド 第 2 巻』の「第 16 章 スレッショルドによる空き領域の管理」にラストチャンス・スレッショルド・プロシージャの例が記載されています。このプロシージャは、dump transaction コマンドを実行して、ログをトランケートします。トランザクション・ログがラストチャンス・スレッショルド・ポイントに達すると、実行中のトランザクションは、領域が使用可能になるまで中断されます。トランザクションが中断されるのは、sybsecurity データベースに対して abort xact when log is full オプションが常に FALSE に設定されているためです。このオプションは変更できません。

trunc log on chkpt オプションを無効にすると、標準の手順で sybsecurity データベースのバックアップとリカバリを実行できますが、リストアされたデータベース内の監査テーブルが、デバイス障害発生時の状況と同期しない場合があることに注意してください。

監査の有効化と無効化

監査を有効または無効にするには、auditing 設定パラメータとともに sp_configure を使用します。構文は次のとおりです。

```
sp_configure "auditing", [0 | 1 ]
```

- 1 は監査を有効にします。
- 0 は監査を無効にします。

たとえば、監査を有効にするには、次のように入力します。

```
sp_configure "auditing", 1
```

注意 監査を有効にしたとき、または無効にしたときに、監査レコードが自動的に生成されます。表 18-5 (640 ページ) のイベント・コード 73 と 74 を参照してください。

単一テーブル監査

Sybase では、運用システムについては単一デバイスでの監査を使用しないよう強くおすすめします。使用する監査テーブルが 1 つだけの場合は、監査データをアーカイブしている間や監査テーブルをトランケートしている間に受信した監査レコードは失われるからです。単一の監査テーブルを使用している場合、これを防ぐ方法はありません。

使用する監査テーブルが 1 つだけの場合は、監査テーブルが満杯になる可能性が高くなります。監査テーブルが満杯になった場合の処置は、設定パラメータ suspend audit when device full の設定によって決まります。suspend audit when device full を on に設定すると、監査プロセスが中断し、監査可能イベントを生成するすべてのユーザ・プロセスも中断します。suspend audit when device full を off に設定すると、監査テーブルはトランケートされ、その監査テーブル内にあったすべての監査レコードが失われます。

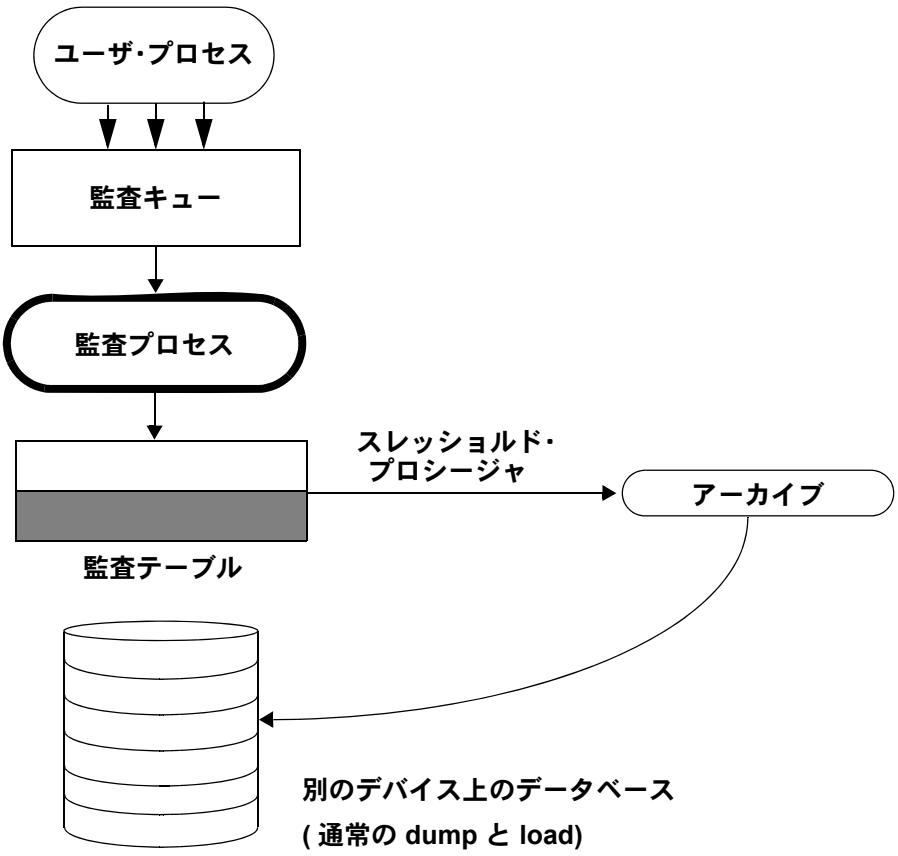
非運用システムの場合、少数の監査レコードの消失であれば、それほど問題はありません。このため、複数の監査テーブルを使用するためのディスク領域を追加できない場合や、使用できる追加デバイスがない場合は、単一テーブルを使用して監査を行います。

単一の監査テーブルを使用するための手順は、複数の監査テーブルを使用する場合に似ていますが、次の例外があります。

- インストール中、監査に使用するシステム・テーブルを 1 つだけ指定する。
- インストール中、監査システム・テーブル用のデバイスを 1 つだけ指定する。
- 監査レコードをアーカイブするために作成するスレッショルド・プロシージャは、複数の監査テーブルを使用する場合のものとは異なる。

図 18-2 は、監査プロセスが単一の監査テーブルを処理する方法を示します。

図 *18-2:* 単一の監査テーブルによる監査



単一テーブル監査の設定と管理

単一の監査テーブルを使用するための手順は、複数の監査テーブルを使用する場合と同じです。詳細については、表 18-1 を参照してください。

単一テーブル監査の監査処理

単一テーブルでの監査の場合は、スレッショルド・プロシージャは次のタスクを実行する必要があります。

- insert コマンドと select コマンドを使用して、満杯に近づいた監査テーブルを別のテーブルにアーカイブする。
- truncate table コマンドを使用して、監査テーブルをトランケートし、新しい監査レコード用の領域を作成する。

監査レコードをアーカイブする前に、監査テーブルと同じカラムを持つアーカイブ・テーブルを作成します。この処理を終えると、スレッショルド・プロシージャで insert と select を使用して監査レコードをアーカイブ・テーブルにコピーすることができます。

次に、単一監査テーブルに使用するサンプル・スレッショルド・プロシージャを示します。

```
create procedure audit_thresh as
/*
** copy the audit records from the audit table to
** the archive table
*/
insert aud_db.sso_user.audit_data
    select * from sysaudits_01
return(0)
go
/*
** truncate the audit table to make room for new
** audit records
*/
truncate table "sysaudits_01"
go
```

スレッショルド・プロシージャを作成したら、そのプロシージャを監査テーブル・セグメントに付加する必要があります。詳細については、「各監査セグメントへのスレッショルド・プロシージャの付加」(619 ページ) を参照してください。

警告！ マルチプロセッサ上では、監査テーブルが満杯になる前にトリガされるスレッショルド・プロシージャがあっても、監査テーブルが満杯になる可能性があります。たとえば、スレッショルド・プロシージャが負荷の重い CPU 上で実行されていて、監査可能なイベントを実行するユーザ・プロセスが負荷の比較的軽い CPU 上で実行されている場合、スレッショルド・プロシージャがトリガする前に、監査テーブルが満杯になる場合があります。設定パラメータ suspend audit when device full は、監査テーブルが満杯になったときの動作を指定します。このパラメータの設定方法については、「デバイスが満杯の場合の監査の中断」(621 ページ) を参照してください。

現在の監査テーブルが満杯になったときに起こる動作

現在の監査テーブルが満杯になると、次の動作が起こります。

1　監査プロセスは、テーブルに次の監査レコードを挿入しようとします。挿入はできないので、監査プロセスは終了します。エラー・メッセージは、エラー・ログに書き込まれます。

2　監査可能イベントをユーザが実行しようとしても、監査を進められないので、そのイベントは終了できません。ユーザ・プロセスは終了します。監査可能イベントを実行しないユーザは、影響を受けません。

3　ログイン監査が有効な場合は、システム・セキュリティ担当者以外は誰もサーバにログインできません。

4　sso_role をアクティブにして実行されるコマンドが監査対象の場合は、システム・セキュリティ担当者はコマンドを実行できません。

現在の監査テーブルが満杯になったときのリカバリの方法

現在の監査デバイスが満杯になって、監査キューも満杯である場合、システム・セキュリティ担当者の操作は監査の対象外となります。この時点から、システム・セキュリティ担当者によって監査可能イベントが実行されると、警告メッセージがエラー・ログ・ファイルに送信されます。このメッセージの内容は、日付と時刻、および監査が行われていないことを知らせる警告で、さらに、ログイン名、event コードなど、通常であれば監査テーブルの extrainfo カラムに保管される情報も含まれます。

現在の監査テーブルが満杯のとき、システム・セキュリティ担当者は、「監査テーブルのアーカイブ」(617 ページ) の説明に従って監査テーブルをアーカイブし、トランケートします。システム管理者が shutdown を実行してサーバを停止してから再起動すると、監査が再開します。

監査システムが異常終了した場合、システム・セキュリティ担当者は、現在の監査テーブルがアーカイブされてトランケートされた後でサーバを停止することができます。通常は、システム管理者だけが shutdown を実行できます。

監査の再起動

エラーが発生したために監査プロセスが強制的に終了された場合は、次のように入力することで sp_audit を手動で再起動できます。

```
sp_audit restart
```

監査プロセスの再起動は、現在実行中の監査がないことを条件として行うことができます。ただし、sp_configure “auditing” 1 の入力によって監査プロセスを有効にする必要があります。

グローバル監査オプションの設定

監査機能をインストールした後は、sp_audit を使用して監査オプションを設定できます。sp_audit の構文は次のとおりです。

sp_audit *option*, *login_name*, *object_name* [,*setting*]

sp_audit にパラメータを付けずに実行すると、すべてのオプションのリストが表示されます。sp_audit の詳細については、『リファレンス・マニュアル：プロシージャ』を参照してください。

注意 サーバの監査がアクティブ化されていないときは、監査は行われません。監査を起動する方法については、「監査の有効化と無効化」(623 ページ) を参照してください。

監査オプション：タイプと要件

sp_audit で *login_name* と *object_name* の各パラメータに指定できる値は、指定する監査オプションのタイプによって異なります。

- グローバル・オプションは、サーバのブートやディスク・コマンド、独自のユーザ定義監査レコードを可能にするかどうかなど、サーバ全体に影響するコマンドに適用されます。グローバル・イベントのオプション設定は、sybsecurity..sysauditoptions システム・テーブルに保存されます。
- データベース固有のオプションはデータベースに適用されます。データベース固有のオプションの例としては、データベースの変更、データベースへのデータのバルク・コピー (bcp in)、データベース内のオブジェクトへのアクセス権の付与および取り消し、データベース内へのオブジェクト作成などがあります。データベース固有のイベントのオプション設定は、master..sysdatabases システム・テーブルに保存されます。
- オブジェクト固有のオプションは、特定のオブジェクトに適用されます。オブジェクト固有のオプションの例としては、特定のテーブルやビューのローの選択、挿入、更新あるいは削除、および特定のトリガやプロシージャの実行などがあります。オブジェクト固有イベントのオプション設定は、関連するデータベース内の sysobjects システム・テーブルに保存されます。
- ユーザ固有のオプションは、特定のユーザあるいはシステム標準の役割に適用されます。ユーザ固有のオプションの例には、テーブルやビューへの特定のユーザのアクセス権や、特定のシステム標準の役割 (sa_role など) がアクティブな状態で実行されるすべてのアクションなどがあります。個々のユーザに関するオプション設定は、master..syslogins に保存されます。システム標準の役割の設定は、master..sysauditoptions に保存されます。

表 18-2 は次のことを示します。

- option の有効な値および各オプションのタイプ － グローバル、データベース固有、オブジェクト固有、あるいはユーザ固有
- 各オプションの、*login_name* パラメータおよび *object_name* パラメータの有効な値
- 監査オプションの設定時に使用するデータベース
- そのオプションを設定したときに監査を受けるコマンドまたはアクセス
- 各オプションの例

すべてのオプションは、デフォルトでオフになっています。

表 *18-2:* 監査オプション、要件および例

オプション (オプションのタイプ)	*login_name*	*object_name*	オプション設定時に使用するデータベース	監査されるコマンドまたはアクセス
adhoc (ユーザ固有)	all	all	すべて	ユーザは sp_addauditrecord を使用できる
	例：`sp_audit "adhoc", "all", "all", "on"` (独自のユーザ定義監査レコードを使用可能にする)			
all (ユーザ固有)	ログイン名または役割	all	すべて	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
	例 `sp_audit "all", "sa_role", "all", "on"` (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)			
alter (データベース固有)	all	監査されるデータベース	すべて	alter database、alter table
	例 `sp_audit @option = "alter", @login_name = "all", @object_name = "master", @setting = "on"` (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)			
bcp (データベース固有)	all	監査されるデータベース	すべて	bcp in
	例 `sp_audit "bcp", "all", "pubs2"` (pubs2 データベースでの bcp の監査のステータスを返す。*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)			
bind (データベース固有)	all	監査されるデータベース	すべて	sp_bindefault、sp_bindmsg、sp_bindrule
	例 `sp_audit "bind", "all", "planning", "off"` (planning データベースの bind 監査をオフにする)			

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
cmdtext (ユーザ固有)	監査されるユーザのログイン名	all	すべて	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)
	例 `sp_audit "cmdtext", "sa", "all", "off"` (データベース所有者の text 監査をオフにする)			
create (データベース固有)	all	監査されるデータベース	すべて	create database、create table、create procedure、create trigger、create rule、create default、sp_addmessage、create view、create index、create function
	注意 create database を監査する場合は、*object_name* に master を指定する。これにより master 内の他のオブジェクトの作成も監査対象となる。 **例** `sp_audit "create", "all", "planning", "pass"` (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。master データベースを指定していないため、create database の監査の現在のステータスは影響を受けない)			
dbaccess (データベース固有)	all	監査されるデータベース	すべて	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
	例 `sp_audit "dbaccess", "all", "project", "on"` (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)			
dbcc (グローバル)	all	all	すべて	パーミッションを必要とするすべての dbcc コマンド
	例 `sp_audit "dbcc", "all", "all", "on"` (dbcc コマンドのすべての実行を監査する)			
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
	例 `sp_audit "delete", "all", "default table", "on"` (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する)			
disk (グローバル)	all	all	すべて	disk init、disk refit、disk reinit、disk mirror、disk unmirror、disk remirror、disk resize
	例 `sp_audit "disk", "all", "all", "on"` (サーバについて、すべてのディスク・アクションを監査する)			

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	すべて	drop database、drop table、drop procedure、drop index、drop trigger、drop rule、drop default、sp_dropmessage、drop view、drop function
	例 `sp_audit "drop", "all", "financial", "fail"` (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)			
dump (データベース固有)	all	監査されるデータベース	すべて	dump database、dump transaction
	例 `sp_audit "dump", "all", "pubs2", "on"` (pubs2 データベース内のダンプ・コマンドを監査する)			
encryption_key (データベース固有)	all	監査されるデータベース	すべて	alter encryption key create encryption key drop encryption key sp_encryption
	例 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	すべて	致命的なエラー、致命的ではないエラー
	例 `sp_audit "errors", "all", "all", "on"` (サーバ全体でのエラーを監査する)			
errorlog	all	all	すべて	sp_errorlog 関数または errorlog_admin 関数
	例 `sp_audit "errorlog", "all", "all", "on"` (「ログの変更」によって新しい Adaptive Server エラー・ログ・ファイルに移動する試みを監査する)			
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)	**実行**
	例 `sp_audit "exec_procedure", "all", "default procedure", "off"` (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)			
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
	例 `sp_audit "exec_trigger", "all", "trig_fix_plan", "fail"` (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)			

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_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	すべて	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
	例 `sp_audit @option="func_dbaccess", @login_name="all", @object_name = "strategy", @setting = "on"` (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)			
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	すべて	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
	例 `sp_audit @option="func_obj_access", @login_name="all", @object_name = "customer", @setting = "on"` (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)			
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	すべて	grant
	例 `sp_audit @option="grant", @login_name="all", @object_name = "planning", @setting = "on"` (planning データベース内のすべての権限の付与を監査する)			
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
	例 `sp_audit "insert", "all", "dpt_101_view", "on"` (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)			
install (データベース固有)	all	監査されるデータベース	すべて	install java
	例 `sp_audit "install", "all", "planning", "on"` (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)			
load (データベース固有)	all	監査されるデータベース	すべて	load database、load transaction
	例 `sp_audit "load", "all", "projects_db", "fail"` (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)			

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
login	all	all	すべて	Adaptive Server へのログイン
(グローバル)	**例** `sp_audit "login", "all", "all", "fail"`			
	(サーバへの失敗したログイン試行をすべて監査する)			
login_locked	all	all	すべて	
(グローバル)	**例** `sp_audit "login_locked", "all", "all", "on"`			
	(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)			
logout	all	all	すべて	Adaptive Server からのログアウト
	例 `sp_audit "logout", "all", "all", "off"`			
	(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)			
mount	all	all	すべて	mount database
(グローバル)	**例** `sp_audit "mount", "all", "all", "on"`			
	(発行されたすべての mount database コマンドを監査する)			
password	all	all	すべて	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
	例 `sp_audit "password", "all", "all", "on"`			
quiesce	all	all	すべて	quiesce database
(グローバル)	**例** `sp_audit "quiesce", "all", "all", "on"`			
	(quiesce database コマンドに対する監査をオンにする)			
reference	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	すべて	create table、alter table
(オブジェクト固有)	**例** `sp_audit "reference", "all", "titles", "off"`			
	(titles テーブルへの参照の作成に対する監査をオフにする)			
remove	all	all	すべて	Java クラスの削除を監査する。
(データベース固有)	**例** `sp_audit "remove", "all", "planning", "on"`			
	(planning データベース内の java クラスの削除を監査する)			
revoke	all	監査されるデータベース	すべて	revoke
(データベース固有)	**例** `sp_audit "revoke", "all", "payments_db", "off"`			
	(payments_db データベース内の revoke の実行の監査をオフにする)			
rpc	all	all	すべて	リモート・プロシージャ・コール (受信と発信の両方)
(グローバル)	**例** `sp_audit "rpc", "all", "all", "on"`			
	(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・コールを監査する)			

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
security (グローバル)	all	all	すべて	サーバ全体のすべてのセキュリティ関連イベント。表 18-5 の “security” オプションを参照。
	例 `sp_audit "security", "all", "all", "on"` (サーバ全体でのセキュリティ関連イベントを監査する)			
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
	例 `sp_audit "select", "all", "customer", "fail"` (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)			
setuser (データベース固有)	all	all	すべて	setuser
	例 `sp_audit "setuser", "all", "projdb", "on"` (projdb データベース内の setuser の実行をすべて監査する)			
table_access (ユーザ固有)	監査されるユーザのログイン名。	all	すべて	テーブル内での select、delete、update または insert によるアクセス
	例 `sp_audit "table_access", "smithson", "all", "on"` ("smithson" というログインによるすべてのテーブル・アクセスを監査する)			
transfer_table (グローバル)	all	all	すべて	サーバワイドなオプション。sysauditoptions には表示されない。
	例 `sp_audit "transfer_table", "tdb1.table1", "all", "on"` (サーバ全体での転送関連イベントを監査する)			
truncate (データベース固有)	all	監査されるデータベース	すべて	truncate table
	例 `sp_audit "truncate", "all", "customer", "on"` (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)			
unbind (データベース固有)	all	監査されるデータベース	すべて	sp_unbindefault、sp_unbindrule、sp_unbindmsg
	例 `sp_audit "unbind", "all", "master", "fail"` (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)			
unmount (グローバル)	all	all	すべて	unmount database
	例 `sp_audit "unmount", "all", "all", "on"` (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)			

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
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
	例 `sp_audit "update", "all", "projects", "on"` (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)			
view_access (ユーザ固有)	監査されるユーザのログイン名	all	すべて	ビューへの select、delete、insert または update
	例 `sp_audit "view_access", "joe", "all", "off"` ("joe" というユーザのビュー監査をオフにする)			

監査オプションの設定の例

company_operations データベース内の projects テーブル、およびそのデータベース内の新しいテーブルすべてに対して失敗したすべての delete を監査する場合を想定します。projects テーブルの監査にはオブジェクト固有の delete オプションを使用し、データベース内の今後作成されるすべてのテーブルの監査には default table オプションを使用します。オブジェクト固有の監査オプションを設定するには、sp_audit を実行する前にそのオブジェクトのデータベースに移動する必要があります。

```
sp_audit "security", "all", "all", "fail"
```

この例では、次のコマンドを実行します。

```
use company_operations
go
sp_audit "delete", "all", "projects", "fail"
go
sp_audit "delete", "all", "default table",
"fail"
go
```

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、次のコマンドを実行します。

```
sp_password 'oldpassword', 'newpassword'
```

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。

```
sp_password '******', '******'
```

これで監査ログにアクセスできる他のユーザにパスワードを見られる心配がなくなります。

現在の監査設定の判別

指定オプションに関する現在の監査設定を判別するには、sp_displayaudit を使用します。構文は次のとおりです。

```
sp_displayaudit [procedure | object | login | database | global |
    default_object | default_procedure [, name]]
```

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

監査証跡へのユーザ指定レコードの追加

sp_addauditrecord を使用すると、ユーザは、監査証跡にコメントを入力できます。構文は次のとおりです。

```
sp_addauditrecord [text] [, db_name] [, obj_name]
    [, owner_name] [, dbid] [, objid]
```

パラメータはすべて省略可能です。

- *text* は、監査テーブル extrainfo に追加するメッセージのテキストです。
- *db_name* は、レコードで参照されるデータベースの名前です。これは、現在の監査テーブルの dbname カラムに挿入されます。
- *obj_name* は、レコードで参照されるオブジェクトの名前です。これは、現在の監査テーブルの objname カラムに挿入されます。
- *owner_name* は、レコードで参照されるオブジェクトの所有者です。これは、現在の監査テーブルの objowner カラムに挿入されます。

- *dbid* は、db_name のデータベース ID 番号を表す整数値です。これは、現在の監査テーブルの dbid カラムに挿入されます。引用符で囲まないでください。
- *objid* は、obj_name のオブジェクト ID 番号を表す整数値です。引用符で囲まないでください。*objid* は、現在の監査テーブルの objid カラムに挿入されます。

sp_addauditrecord は次の場合に使用できます。

- 実行するユーザが、sp_addauditrecord　に対する実行パーミッションを持っている。
- 監査設定パラメータが sp_configure によってアクティブ化されている。
- adhoc 監査オプションが sp_audit によって有効化されている。

デフォルトでは、システム・セキュリティ担当者と sybsecurity のデータベース所有者のみが sp_addauditrecord を使用できます。その実行パーミッションは別のユーザに付与できます。

ユーザ定義監査レコードの追加例

次の例では、現在の監査テーブルにレコードを追加します。テキスト部分は現在の監査テーブルの extrainfo カラムに、“corporate” は dbname カラムに、“payroll” は objname カラムに、“dbo” は objowner カラムに、“10” は dbid カラムに、“1004738270” は objid カラムにそれぞれ挿入されます。

```
sp_addauditrecord "I gave A. Smith permission to view the
payroll table in the corporate database.This permission was in
effect from 3:10 to 3:30 pm on 9/22/92.", "corporate",
"payroll", "dbo", 10, 1004738270
```

次の例は、現在の監査テーブルの extrainfo カラムと dbname カラムにだけ情報を挿入します。

```
sp_addauditrecord @text="I am disabling auditing briefly while
we reconfigure the system", @db_name="corporate"
```

監査証跡のクエリ

監査証跡を問い合わせるには、SQL を使用して、監査データを選択および要約します。「監査証跡の管理の設定」(616 ページ) で説明している手順に従った場合は、監査データは別のデータベース内の 1 つまたは複数のテーブルに自動的にアーカイブされます。たとえば、監査データが、audit_db データベースの audit_data というテーブル内にあるとします。この場合、“bob” によって 1993 年 7 月 5 日に実行されたタスクの監査レコードを選択するには、次のコマンドを実行します。

```
use audit_db
go
select * from audit_data
   where loginname = "bob"
   and eventtime like "Jul 5% 93"
go
```

次のコマンドでは、システム・セキュリティ担当者の役割がアクティブなユーザによって、pubs2 データベースで実行されたコマンドの監査レコードを要求します。

```
select * from audit_data
   where extrainfo like "%sso_role%"
   and dbname = "pubs2"
go
```

次のコマンドでは、すべてのテーブル・トランケーション (イベント 64) の監査レコードを要求します。

```
select * from audit_data
   where event = 64
go
```

監査イベントの名前を使用して監査証跡を問い合わせるには、audit_event_name 関数を使用します。たとえば、すべてのデータベース作成イベントに対する監査レコードを要求するには、次のように入力します。

```
select * from audit_data where audit_event_name(event)
   = “Create Database”
go
```

監査テーブルの概要

システム監査テーブルにアクセスできるのはシステム・セキュリティ担当者だけで、システム・セキュリティ担当者は SQL コマンドを実行してテーブルを読み込むことができます。システム監査テーブルに対して使用できるコマンドは、select と truncate だけです。

表 18-3 は、すべての監査テーブルにあるカラムの説明です。

表 *18-3:* 各監査テーブル内のカラム

カラム名	データ型	説明
event	smallint	監査されるイベントのタイプ。表 18-5 (640 ページ) を参照してください。
eventmod	smallint	監査されるイベントに関する詳細。該当するイベントがパーミッション検査に合格したかどうかを示す。値は次のとおり。 • 0 = このイベントの修飾子はない。 • 1 = イベントがパーミッションの検査に成功した。 • 2 = イベントがパーミッションの検査に失敗した。
spid	smallint	監査レコードの書き込みが発生したプロセスのサーバ・プロセス ID。
eventtime	datetime	監査イベントが起こった日付と時刻。
sequence	smallint	単一イベント内のレコードのシーケンス番号。一部のイベントは、複数の監査レコードを必要とする。
suid	smallint	監査イベントを実行したユーザのサーバ・ログイン ID。
dbid	int null	監査されるイベントが発生したデータベースの ID、または、オブジェクト、ストアド・プロシージャ、トリガが存在するデータベースの ID (イベントのタイプによる)。
objid	int null	アクセスされたオブジェクト、ストアド・プロシージャ、またはトリガの ID。
xactid	binary(6) null	監査イベントを含むトランザクション ID。マルチデータベース・トランザクションの場合は、トランザクションが開始したデータベースからのトランザクション ID。
loginname	varchar(30) null	suid に対応するログイン名。
dbname	varchar(30) null	dbid に対応するデータベース名。
objname	varchar(30) null	objid に対応するオブジェクト名。
objowner	varchar(30) null	objid の所有者名。
extrainfo	varchar(255) null	監査イベントについての追加情報。このカラムに格納される一連の項目は、セミコロンで区切られている。詳細については、「extrainfo カラムの読み込み」(639 ページ) を参照してください。
nodeid	tinyint	イベントが発生したクラスタ内のサーバの nodeid

extrainfo カラムの読み込み

extrainfo カラムには、一連のデータがセミコロンで区切られて格納されています。このデータは、次のカテゴリから構成されます。

表 *18-4: extrainfo* カラム内の情報

位置	カテゴリ	説明
1	役割	アクティブな役割をブランクで区切ったリスト。
2	キーワードまたはオプション	イベントに使用されたキーワードまたはオプションの名前。たとえば、alter table コマンドでは、add column オプションや drop constraint オプションなどが使用される。複数のキーワードまたはオプションの場合は、カンマで区切られる。
3	以前の値	イベントによって値が更新された場合は、更新される前の値がこの項目に格納される。
4	現在の値	イベントによって値が更新された場合は、新しい値がこの項目に格納される。

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	代理権限情報	set proxy が有効なときにイベントが発生した場合は、元のログイン名が格納される。
7	プリンシパル名	ユーザのログインがセキュア・デフォルト・ログインであり、ユーザが統一化ログインを介して Adaptive Server にログインした場合に、基本となるセキュリティ・メカニズムのプリンシパル名が格納される。セキュア・デフォルト・ログインが使用されていない場合、この項目の値は NULL。

次の例は、監査設定パラメータを変更するイベントの extrainfo カラムを示します。

```
sso_role;suspend audit when device full;1;0;;ralph;
```

このエントリは、システム・セキュリティ担当者が、設定パラメータ suspend audit when device full を 1 から 0 に変更したことを示します。このエントリに “Other information” はありません。6 番目のカテゴリは、ユーザ “ralph” が代理ログインによって操作していたことを示します。プリンシパル名はありません。

監査レコードの他のフィールドには、他の関連情報が格納されます。たとえば、サーバ・ユーザ ID (suid) とログイン名 (loginname) もレコードに含まれています。

表 18-5 は、event カラムに表示される値を sp_audit のオプション順にリストにしたものです。「extrainfo 出力の情報」の欄では、監査テーブルの extrainfo カラムに表示される情報を、表 18-4 に示すカテゴリに基づいて説明しています。

表 *18-5: event* カラムと *extrainfo* カラムの値

監査オプション	監査されるコマンドまたはアクセス	イベント	extrainfo の情報
(オプションによって制御されるのではなく、自動的に監査されるイベント)	監査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	—
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 パラメータの値が挿入される

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	alter database	2	サブコマンド・キーワード： alter maxhold alter size inmemory
	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 暗号化カラムが 1 つ以上追加される場合、extrainfo キーワードに次のものが含まれる。add/drop/modify column *column1*/*keyname1*, [,*column2*/*keyname2*] ここで、*keyname* はキーの完全修飾名です。
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 procedure	11	—
	create rule	13	—
	create table	10	暗号化カラムでは、extrainfo にはカラム名とキー名が含まれます。 EK *column1*/*keyname1*[,*column2 keyname2*] このとき、EK は、後続の情報が暗号化キーを参照することを示すプレフィクスです。また、*keyname* はキーの完全修飾名です。
	create trigger	12	—
	create view	16	—
	create index	104	その他の情報：インデックス名
	create function	97	—
	sp_addmessage	15	その他の情報：メッセージ番号
dbaccess	すべてのユーザによるデータベースへのあらゆるアクセス	17	キーワードまたはオプション： use cmd outside reference

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
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
	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 ******** ********`

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 encryption key	107	キーワードの内容は次のとおりです。 algorithm name-bitlength/IV [random|NULL]/pad [random |NULL] user/system 例： `AES-128/IV RANDOM/PAD NULL USER`
	alter encryption key	108	`default/not default`
	drop encryption key	109	
	AEK modify encryption	118	modify encryption with user passwd | for user *username* {with login passwd | with user passwd | with *keyvalue*} [for recovery *keyvalue* は、alter encryption key modify encryption の複写についてのみ表示されます。たとえば、ユーザ “stephen” がそのキー・コピーを変更すると、次の情報が保存されます。 `MODIFY ENCRYPTION for user stephen WITH USER PASSWD`
	AEK add encryption	119	add encryption for user *user_name* for login association | recovery|with keyvalue] *keyvalue* は、alter encryption key add encryption の複写についてのみ表示されます。
	alter encryption key drop encryption	120	drop encryption [for recovery | for user *user_name* 『暗号化カラム・ユーザーズ・ガイド』を参照してください。
	alter encryption key modify owner	121	modify owner [new owner *user_name*] 『暗号化カラム・ユーザーズ・ガイド』を参照してください。
	alter encryption key recover key	122	recovery key [with *key_value*] with *keyvalue* は、alter encryption key の複写時にのみ使用されます。 『暗号化カラム・ユーザーズ・ガイド』を参照してください。
errorlog	errorlog 関数または errorlog_admin 関数	127	errorlog_admin に渡されたパラメータは、サブコマンドの特定のために記録されます：errorlog_admin (param1, param2,...)
errors	致命的なエラー	36	その他の情報：*Error number.Severity.State*
	致命的ではないエラー	37	その他の情報：*Error number.Severity.State*
exec_procedure	プロシージャの実行	38	その他の情報：すべての入力パラメータ

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_trigger	トリガの実行	39	—
func_obj_access、func_dbaccess	Transact-SQL 関数を介したオブジェクトおよびデータベースへのアクセス (関数を監査するには、sa_role について監査を有効にする必要があります)。	86	—
grant	grant	40	—
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
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	—
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
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： -d*masterdevicename* -i*interfaces file path* -S*servername* -e*errorfilename*
	sp_webservices	111	キーワードまたはオプション：単一の Web サービスを配備する場合は deploy、すべての Web サービスを配備する場合は deploy_all
	sp_webservices	111	キーワードまたはオプション：単一の Web サービスの配備を解除する場合は undeploy、すべての Web サービスの配備を解除する場合は undeploy_all
	サーバの停止	51	キーワードまたはオプション：shutdown
	set proxy または set session authorization	88	以前の値：以前の suid 現在の値：新しい suid
	sp_configure	82	キーワードまたはオプション：SETCONFIG その他の情報： • パラメータが設定される場合は、設定パラメータの数 • 設定ファイルを使用してパラメータが設定される場合は、その設定ファイルの名前
	sp_ssladmin 管理の有効化	99	証明書を追加する場合は、SSL_ADMIN addcert を含むキーワード
	監査テーブルへのアクセス	61	—
	create login、drop login	103	キーワードまたはオプション：create login、drop login
	create、drop、alter、grant、revoke role	85	キーワードまたはオプション：create、drop、alter、grant、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 Adaptive Server の起動	95	その他の情報として、'Unlocking admin account' が保存される

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
	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ています。
	システムまたは sp_passwordpolicy による、ネットワーク・パスワードの暗号化のための非対称キーペアの再生成	117	extrainfo の情報
select	テーブルからの select	62	キーワードまたはオプション： select into select readtext
	ビューからの select	63	キーワードまたはオプション： select into select readtext
setuser	setuser	84	その他の情報：設定されたユーザの名前
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
	ビューの update	71	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
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 18-6 は、event カラムに表示される値を監査イベント順にリストにしたものです。

表 *18-6:* 監査イベント値

監査イベント ID	コマンド名	監査イベント ID	コマンド名
1	ad hoc audit record	62	select table
2	alter database	63	select view
3	alter table	64	truncate table
4	bcp in	65	予約済み
5	予約済み	66	予約済み
6	bind default	67	unbind default
7	bind message	68	unbind rule
8	bind rule	69	unbind message
9	create database	70	update table
10	create table	71	update view
11	create procedure	72	予約済み
12	create trigger	73	監査の有効化
13	create rule	74	監査の無効化
14	create default	75	予約済み
15	create message	76	SSO が変更したパスワード
16	create view	77	予約済み
17	access to database	78	予約済み
18	delete table	79	予約済み
19	delete view	80	役割チェックの実行
20	disk init	81	dbcc
21	disk refit	82	config
22	disk reinit	83	online database
23	disk mirror	84	setuser コマンド
24	disk unmirror	85	UDR コマンド
25	disk remirror	86	組み込み関数
26	drop database	87	ディスクの解放

監査イベント ID	コマンド名	監査イベント ID	コマンド名
27	drop table	88	set SSA コマンド
28	drop procedure	89	kill コマンドまたは terminate コマンド
29	drop trigger	90	connect
30	drop rule	91	reference
31	drop default	92	コマンド・テキスト
32	drop message	93	JCS install コマンド
33	drop view	94	JCS remove コマンド
34	dump database	95	管理者アカウントのロック解除
35	dump transaction	96	quiesce database
36	致命的なエラー	97	create SQLJ 関数
37	致命的ではないエラー	98	drop SQLJ 関数
38	ストアド・プロシージャの実行	99	SSL 管理
39	トリガの実行	100	disk resize
40	grant	101	mount database
41	insert table	102	unmount database
42	insert view	103	login コマンド
43	load database	104	create index
44	load transaction	105	drop index
45	ログイン	106	sp_encryption (暗号化された列の管理)
46	logout	107	create encryption key
47	revoke	108	Alter Encryption Key as/not default
48	rpc in	109	drop encryption key
49	rpc out	110 111	deploy user-defined web services undeploy user defined web services
50	server boot	112	ログインがロックされている
51	サーバのシャットダウン	113	quiesce hold security
52	予約済み	114	quiesce release
53	予約済み	115	パスワード管理
54	予約済み	116	create manifest file
55	役割のオンとオフ	117	regenerate keypair
56	予約済み	118	alter encryptin key modify encryption
57	予約済み	119	alter encryption key add encryption

監査イベント ID	コマンド名	監査イベント ID	コマンド名
58	予約済み	120	alter encryption key drop encryption
59	予約済み	121	alter encryption key modify owner
60	予約済み	122	alter encryption key for key recovery
61	監査テーブルへのアクセス	123	LDAP ステータス変更
		127	エラー・ログの管理
		136	transfer table

失敗したログイン試行のモニタリング

ログイン試行の失敗回数が所定の限度を超えたためにログイン・アカウントがロックされると、監査オプションの login_locked と Locked Login (値 112) イベントが記録されます。このイベントは監査オプションの login_locked が設定されると有効になります。login_locked を設定するには、次のように入力します。

```
sp_audit "login_locked","all","all","ON"
```

監査テーブルが満杯でイベントを記録できない場合は、その情報がエラー・ログに記録されます。

ホスト名とネットワークの IP アドレスが監査レコードに記録されます。監査ログを使用して Locked Login イベント (数値 112) をモニタリングすると、ログイン・アカウントに対する攻撃の識別に役立ちます。

ログイン失敗の監査

クライアント・アプリケーションはさまざまな理由でログインに失敗することがありますが、Adaptive Server では、ログイン失敗に関する詳細な情報を提供しません。これは、パスワードの解読や Adaptive Server の認証メカニズムの侵害を意図している悪意のあるユーザに情報を与えることを避けるためです。

ただし、詳細情報は、システム管理者にとっては Adaptive Server の管理上の問題や設定上の問題を診断するために、セキュリティ担当者にとってはセキュリティの侵害を調査するために役に立ちます。

次のように指定することで、すべてのログイン失敗を監査できます。

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

情報の不正使用を防止するために、SSO 役割を付与されたユーザだけが、この機密情報を含む監査証跡情報にアクセスできます。

Adaptive Server は、次の条件に該当するログイン失敗を監査します。

- Windows サービスとして起動された Adaptive Server で、Sybase SQLServer サービスが一時停止された (たとえば Microsoft Management Console for Services によって停止された)。
- リモート・サーバがサーバ対サーバ RPC 用のサイト・ハンドラを確立しようとしたが、リソース不足のため (またはその他の理由で) サイト・ハンドラを初期化できなかった。
- Windows 版の Adaptive Server を trusted ログインまたは統一化ログインを設定して使用しようとしたが、指定されたユーザが信頼された管理者ではなかった (認証できなかった)。
- Adaptive Server が、クライアントによって要求された SQL インタフェースをサポートしていない。
- Adaptive Server がシングルユーザ・モードで稼動してるときにユーザがログインしようとした。シングルユーザ・モードでは、sa_role が付与されているユーザが 1 人だけ Adaptive Server にアクセスできます。sa_role を持っているユーザであっても、追加ログインはできません。
- master データベース内の syslogins テーブルが開かない。これは、master データベースに内部エラーがあることを示します。
- クライアントがリモート・ログインしようとしたが、sysremotelogins が開かない。または、指定されたユーザ・アカウント用のエントリがなく、ゲスト・アカウントも存在しない。
- クライアントがリモート・ログインしようとしたが、指定されたユーザの sysremotelogins 内のエントリがローカル・アカウントを参照しているにもかかわらず、参照先のローカル・アカウントが存在しない。
- クライアント・プログラムがセキュリティ・セッション (Kerberos 認証など) を要求しているが、次の理由でセキュリティ・セッションを確立できない。
  - Adaptive Server のセキュリティ・サブシステムが起動時に初期化されなかった。
  - 構造体に割り当てるメモリ・リソースが不足している。
  - 認証のネゴシエーションが失敗した。
- 指定されたユーザに対して実行される認証メカニズムが見つからない。
- 指定されたパスワードが正しくなかった。
- 指定されたログインで必要なエントリが syslogins に含まれていない。
- ログイン・アカウントがロックされている。
- Adaptive Server のユーザ接続数が制限値に達した。

- unified login required パラメータが設定されているが、適切なセキュリティ・サブシステムによってログインが認証されていない。
- Adaptive Server のネットワーク・バッファを使用できない、または要求されたパケット・サイズが無効である。
- クライアント・アプリケーションがホスト・ベースの通信ソケット接続を要求しているが、ホスト・ベースの通信バッファ用にメモリ・リソースを使用できない。
- シャットダウンが進行中だが、指定されたユーザは SA 役割を持っていない。
- Adaptive Server がログイン用のデフォルト・データベースを開くことができなかった。かつ、このログインには master データベースへのアクセス権がない。
- クライアントは高可用性ログイン・フェールオーバを要求しているが、高可用性サブシステムがこのログインに対して高可用性セッションを確立していない、またはフェールオーバが完了するまでログインが待機できない。
- クライアントは高可用性ログイン設定を要求しているが、高可用性サブシステムがセッションを確立できない、または高可用性セッションのための TDS プロトコル・ネゴシエーションを完了できない。
- Adaptive Server が、ログインに対して tempdb を設定できない。
- TDS ログイン・プロトコル・エラーが検出された。

第 19 章

データの機密保持

この章では、すべてのデータを保護し、機密性を保持するための Adaptive Server の設定方法について説明します。

Adaptive Server における SSL (Secure Sockets Layer)

Adaptive Server Enterprise セキュリティ・サービスは、現在 SSL (Secure Sockets Layer) セッションベースのセキュリティをサポートしています。**SSL** は、クレジット・カード番号、株式売買、銀行取引などの機密情報を、インターネット上で安全に転送するための標準です。

このマニュアルでは、パブリック・キー暗号法については詳しく説明しませんが、SSL によってインターネット通信チャネルの安全性が保証される仕組みを理解できるように、基本的なことについては説明します。このマニュアルは、パブリック・キー暗号法の全般的なガイドではありません。

Adaptive Server SSL 機能の実装は、ユーザ・サイトのセキュリティ・ポリシーとニーズを熟知し、SSL およびパブリック・キー暗号法について全般的な知識のあるシステム・セキュリティ担当者がいることを前提としています。

インターネット通信の概要

TCP/IP は、クライアント／サーバ・コンピューティングで使用されるプライマリ・トランスポート・プロトコルであり、インターネットへのデータ転送を制御するプロトコルです。TCP/IP では、送信側から受信側へデータが転送されるときに、いくつもの中間コンピュータを経由します。複数のコンピュータを経由することによって、通信システムの中に安全性の低いリンクが生じ、データの改ざん、盗難、盗聴、なりすましなどを受けやすくなります。

パブリック・キー暗号法

「パブリック・キー暗号法」とは、機密を要するデータをインターネットでの転送中に保護するために開発され、実装されている、さまざまなメカニズムの総称です。パブリック・キー暗号法は、暗号化、キー交換、デジタル署名、デジタル証明書から構成されます。

復号化

暗号化のプロセスでは、暗号化アルゴリズムを使用して情報をコード化し、その情報を目的の受信者以外の者から保護します。暗号化に使用するキーには、次の 2 種類があります。

- 対称キー暗号化では、メッセージの暗号化と復号化に同じアルゴリズム (キー) を使用します。この暗号化方式では、簡単に解読できる単純なキーを使用しているため、最低限のセキュリティしか保証されません。しかし、対称キーによる暗号化の場合は、メッセージの暗号化と復号化に必要な計算の量が最小限で済むため、データ転送が高速になります。
- パブリック・キー／プライベート・キー (非対称キー) 暗号化では、公開コンポーネントと秘密コンポーネントから成る一対のキーを使用してメッセージの暗号化と復号化を行います。通常、送信者はプライベート・キーを使用してメッセージを暗号化し、受信者は送信者のパブリック・キーを使用してメッセージを復号化しますが、この組み合わせは異なる場合もあります。送信者が受信者のパブリック・キーを使ってメッセージを暗号化し、受信者が受信者自身のプライベート・キーを使用してメッセージを復号化することも可能です。

  パブリック・キーとプライベート・キーを作成するときに使用するアルゴリズムは複雑なので、解読するのは容易ではありません。しかし、パブリック・キー／プライベート・キー暗号化では、より多くの計算が必要となり、接続を介して送られるデータの量も増えるので、データ転送が遅くなります。

キー交換

安全性を損なうことなく、計算によるオーバヘッドを減らしてトランザクションを高速化するには、対称キー暗号化とパブリック・キー／プライベート・キー暗号化の両方を組み合わせて使用します。この方法を、キー交換と呼びます。

データ量が多い場合は、対称キーを使用して元のメッセージを暗号化します。次に、送信者は、送信者自身のプライベート・キーまたは受信者のパブリック・キーを使用して、対称キーを暗号化します。暗号化されたメッセージと暗号化された対称キーの両方が受信者に送信されます。メッセージを暗号化するときにはパブリック・キーまたはプライベート・キーを使用しますが、そのときに使用しなかった方のキーを使用して、受信者は対称キーを復号化します。キーの交換が終了すると、受信者は対称キーを使用してメッセージを復号化します。

デジタル署名

デジタル署名は、不正な変更を検出したり否認を防止したりするために使用されます。テキスト／メッセージからユニークな固定長の文字列になった数字を生成する数値アルゴリズムを使用して、デジタル署名は作成されます。この生成された数値はハッシュまたはメッセージ・ダイジェストと呼ばれます。

メッセージの整合性を保証するために、メッセージ・ダイジェストは署名者のプライベート・キーで暗号化され、ハッシュ・アルゴリズムについての情報とともに受信者に送信されます。受信者は、署名者のパブリック・キーを使用してメッセージを復号化します。また、この処理では、元のメッセージ・ダイジェストも再生成されます。これらのダイジェストが一致すれば、メッセージは損なわれておらず、改ざんされてもいないことになります。一致しない場合は、転送中にデータが修正されたか、改ざん者によりデータが署名されたことになります。

さらに、デジタル署名によって「否認防止」が可能になります。つまり、送信者は、自身のプライベート・キーでメッセージを暗号化するので、メッセージを送ったことを否定 (否認) できないことになります。ただし、盗難や解読によってプライベート・キーの機密性が損なわれると、デジタル署名は否認防止に役立ちません。

デジタル証明書

デジタル証明書は一種のパスポートです。証明書がユーザに割り当てられると、認証局は、システムにおけるユーザのあらゆる ID 情報を持つことになります。パスポートと同様に、証明書は、あるエンティティ (サーバ、ルータ、Web サイトなど) の身元を他者に対して確認するために使用されます。

Adaptive Server は次の 2 つのタイプの証明書を使用します。

- サーバ証明書 － サーバ証明書は、それを保有しているサーバを認証します。証明書は、信頼された第三者の CA (認証局) によって発行されます。CA は、証明書の保有者の身元を検証し、保有者のパブリック・キーなどの ID 情報を、デジタル証明書に埋め込みます。証明書には、発行元 CA のデジタル署名が含まれています。これによって、証明書データの整合性が確認され、証明書を使用できるようになります。
- 認証局証明書 (信頼されたルート証明書とも呼ばれます) － サーバの起動時にロードされる、信頼された認証局のリストです。認証局証明書は、RPC (リモート・プロシージャ・コール) の間などサーバがクライアントとして機能するときに、サーバによって使用されます。Adaptive Server は、自身の認証局の信頼されたルート証明書を起動時にロードします。Adaptive Server は、RPC を実行するためにリモート・サーバに接続するときに、リモート・サーバの証明書に署名した CA が、Adaptive Server 自身の CA の信頼されたルート・ファイルにある「信頼された」CA かどうかを検証します。信頼された CA でない場合は、接続が許可されません。

証明書は一定期間有効で、認証局は、セキュリティ侵害が生じたときなどさまざまな理由で証明書を無効にすることができます。セッション中に証明書が無効になった場合、そのセッション接続は継続します。後続のログイン試行は失敗します。同様に、証明書の有効期限が切れたときも、ログイン試行は失敗します。

これらのメカニズムの組み合わせにより、インターネットを介して送信されるデータを盗聴や改ざんから守ります。また、なりすまし攻撃からもユーザを保護します。なりすまし攻撃には、あるエンティティが別のエンティティの振りをする (スプーフィング) ものや、組織または個人が、機密情報の入手という本当の目的を隠して別の目的を偽るもの (虚偽の陳述) があります。

SSL の概要

SSL は、ワイヤ・レベルまたはソケット・レベルで暗号化されたデータを、保護されたネットワーク接続を介して送信するための業界標準です。

サーバとクライアントは何度か I/O を交換し、安全な暗号化セッションをネゴシエートして合意してから、SSL 接続が確立されます。これは、SSL ハンドシェイクと呼ばれています。

SSL ハンドシェイク

クライアントが接続を要求すると、SSL が有効化されているサーバは、その身元を証明する証明書を提示してから、データ転送を行います。基本的に、ハンドシェイクは次の手順から成り立っています。

- クライアントがサーバに接続要求を送信します。要求には、クライアントがサポートしている SSL (または TLS: Transport Layer Security) オプションが含まれています。
- サーバは、自身の証明書と、サポートされている暗号スイートのリストを返す。このリストには、SSL/TLS サポート・オプション、キー交換で使用するアルゴリズム、デジタル署名が含まれます。
- クライアントとサーバの両者が 1 つの CipherSuite 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 (http://www.ietf.org) を参照してください。

Adaptive Server がサポートする暗号スイートのリストについては、「暗号スイート」(665 ページ) を参照してください。

Adaptive Server での SSL

Adaptive Server が SSL を実装したことにより、いくつかのレベルでのセキュリティが可能になりました。

- サーバが自身を認証し (ユーザの交信対象のサーバであることを証明する)、データ転送を行う前に、暗号化された SSL セッションを開始する。
- SSL セッションが確立すると、接続を要求するクライアントは暗号化された安全な接続を介してユーザ名とパスワードを送信できる。
- サーバ証明書の電子署名を比較することにより、クライアントが受信したデータが、本来の受信者に到達するまでに修正されたかどうかを判断できる。

ほとんどのプラットフォームで、Adaptive Server は Certicom の SSL Plus(TM) ライブラリ API を使用しています。ただし、Windows Opteron X64 では、Adaptive Server は SSL プロバイダとして OpenSSL を使用しています。

SSL フィルタ

interfaces ファイル、Windows レジストリ、LDAP サービスなどの Adaptive Server のディレクトリ・サービスは、サーバ・アドレスとポート番号を定義し、クライアント接続に使用するセキュリティ・プロトコルを決定します。Adaptive Server では、SSL プロトコルはフィルタとして実装され、ディレクトリ・サービスの master 行と query 行に追加されます。

Adaptive Server が接続を受け付けるアドレスとポート番号は、単一のサーバで複数のネットワーク・プロトコルとセキュリティ・プロトコルを有効にできるように設定することが可能です。サーバ接続の属性は、LDAP などのディレクトリ・サービス、または従来の Sybase の *interfaces* ファイルで指定されます。「サーバ・ディレクトリ・エントリの作成」(662 ページ) を参照してください。

SSL フィルタを使用して *interfaces* ファイルの master エントリまたは query エントリに接続するには、その接続で SSL プロトコルをサポートしている必要があります。SSL 接続を受け付け、別の接続では暗号化されないクリア・テキストを受け付けるようにサーバを設定することも、他のセキュリティ・メカニズムを使用するように設定することもできます。

たとえば、SSL ベースの接続とクリア・テキストの接続の両方をサポートする UNIX の *interfaces* ファイルは、次のようになります。

```
SYBSRV1
master tcp ether myhostname myport1 ssl
query    tcp ether myhostname myport1 ssl
master tcp ether myhostname myport2
```

SSL フィルタは、*interfaces* ファイル (Windows では *sql.ini*) の SECMECH (セキュリティ・メカニズム) 行で定義される DCE や Kerberos などのセキュリティ・メカニズムとは別のものです。

証明書による認証

SSL プロトコルは、暗号化セッションを有効にするために、サーバ証明書によるサーバ認証を要求します。同様に、Adaptive Server が RPC の実行時にクライアントとして機能しているときには、サーバ証明書を検証するためにクライアント接続がアクセスできる、信頼された認証局のレポジトリが必要になります。

サーバ証明書

それぞれの Adaptive Server には、起動時にロードされる専用のサーバ証明書ファイルが必要です。証明書ファイルのデフォルトのロケーションは次のとおりです。*servername* は、起動時にコマンド・ラインで-s フラグを使用して、または環境変数 *$DSLISTEN* を使用して指定される Adaptive Server の名前です。

UNIX　*$SYBASE/$SYBASE_ASE/certificates/servername.crt*

Windows　*%SYBASE%¥%SYBASE_ASE%¥certificates¥servername.crt*

サーバ証明書ファイルは、サーバ証明書と、そのサーバ証明書用の暗号化されたプライベート・キーを含む、コード化されたデータから構成されています。

また、sp_ssladmin を使用して、サーバ証明書ファイルのロケーションを指定することもできます。

注意 クライアントが正しく接続できるようにするには、証明書内の共通名が *interfaces* ファイル内の Adaptive Server 名と一致している必要があります。

認証局の信頼されたルート証明書

信頼された認証局のリストは、Adaptive Server の起動時に、信頼されたルート・ファイルからロードされます。信頼されたルート・ファイルは、フォーマットは証明書ファイルに似ていますが、Adaptive Server が認識する認証局の証明書が格納されている点が異なります。信頼されたルート・ファイルは次のロケーションにあり、ローカルの Adaptive Server からアクセスできます。*servername* はサーバ名です。

- UNIX − *$SYBASE/$SYBASE_ASE/certificates/servername.txt*
- Windows − *%SYBASE%¥%SYBASE_ASE¥certificates¥servername.txt*

信頼されたルート・ファイルが使用されるのは、RPC や CIS (コンポーネント総合サービス) 接続の実行時など、Adaptive Server がクライアントとして機能しているときだけです。

Adaptive Server が受け付ける認証局をシステム・セキュリティ担当者が追加および削除するには、一般的な ASCII テキスト・エディタを使用します。

警告！ Adaptive Server 内部では、システム・セキュリティ担当者の役割 (sso_role) を使用して、セキュリティに関係するオブジェクトに対するアクセスや実行を制限してください。

Adaptive Server には、証明書要求を生成するツールや証明書を認可するためのツールがあります。「Adaptive Server ツールを使用した証明書の要求と認可」(661 ページ) を参照してください。

接続タイプ

この項では、クライアントとサーバの間のさまざまな接続について説明します。

クライアントから Adaptive Server へのログイン

既存のクライアント接続が確立されるのと同じように、Open Client アプリケーションは Adaptive Server へのソケット接続を確立します。ネットワーク・トランスポート・レベルの接続コールがクライアント側で完了し、承認コールがサーバ側で完了すると、ソケット上で SSL ハンドシェイクが行われ、その後でユーザ・データが送信されます。

サーバ間リモート・プロシージャ・コール (RPC)

Adaptive Server が RPC を実行するために他のサーバへのソケット接続を確立する方法は、既存の RPC 接続の確立方法と同じです。ネットワーク・トランスポート・レベルの接続コールが完了して、ソケット上で SSL ハンドシェイクが行われた後で、ユーザ・データが送信されます。サーバ間のソケット接続が既に確立している場合は、既存のソケット接続とセキュリティ・コンテキストが再使用されます。

Adaptive Server は、RPC の実行時にクライアントとして機能しているときは、接続中にリモート・サーバの証明書を要求します。Adaptive Server は、リモート・サーバの証明書に署名した認証局が信頼できることを確認します。つまり、信頼されたルート・ファイルにある、自身の信頼された認証局のリストにあることを確認します。また、サーバ証明書内の共通名が、接続の確立時に使用した共通名と一致していることを確認します。

コンパニオン・サーバと SSL

コンパニオン・サーバを使用してフェールオーバを行うように Adaptive Server を設定できます。プライマリ・サーバとセカンダリ・サーバの両方で、SSL と RPC の設定が同じであるように設定してください。接続がフェールオーバまたはフェールバックされるとき、接続とともにセキュリティ・セッションが再度確立されます。

Open Client 接続

コンポーネント統合サービス、RepAgent、分散トランザクション管理、および Adaptive Server の他のモジュールは、Client Library を使用して Adaptive Server 以外のサーバとの接続を確立します。リモート・サーバはその証明書によって認証されます。リモート・サーバは、ユーザ名とパスワードを使用して、RPC を実行するための Adaptive Server クライアント接続を認証します。

SSL の有効化

Adaptive Server は、interfaces ファイル (Windows では *sql.ini*) に基づいて、各ポートで使用するセキュリティ・サービスを判断します。

❖ SSL の有効化

1　サーバの証明書を生成します。

2　信頼されたルート・ファイルを作成します。

3　sp_configure を使用して、SSL を有効にします。コマンド・プロンプトで、次のように入力してください。

```
sp_configure "enable ssl", 1
```

- 1 − 起動時に SSL サブシステムが有効になり、メモリが割り当てられます。ネットワーク上で送受信されるデータは SSL によってワイヤレベルで暗号化されます。
- 0 (デフォルト) − SSL を無効にします。これはデフォルトの設定です。

4　SSL フィルタを *interfaces* ファイルに追加します。「サーバ・ディレクトリ・エントリの作成」(662 ページ) を参照してください。

5 sp_ssladmin を使用して、証明書ファイルに証明書を追加します。「証明書の管理」(663 ページ) を参照してください。

6 Adaptive Server を停止して再起動します。

注意 第三者の証明書を要求、認証、変換するには、『ユーティリティ・ガイド』の certauth、certreq、certpk12 の各ツールの説明を参照してください。

DCE、Kerberos、NTLAN などの他のセキュリティ・サービスとは異なり、SSL は、Open Client/Open Server 設定ファイル *libtcl.cfg* の “Security” セクションにも、*objectid.dat* 内のオブジェクトにも依存しません。

システム管理者は、物理メモリの総量を計画するときに、SSL で使用するメモリを考慮する必要があります。Adaptive Server で SSL 接続を行う場合、接続ごとに約 40 K のメモリが必要になります (接続にはユーザ接続、リモート・サーバ、ネットワーク・リスナを含む)。メモリは、メモリ・プール内で予約され事前に割り付けられ、Adaptive Server ライブラリと SSL Plus ライブラリにより必要に応じて内部で使用されます。

証明書の取得

システム・セキュリティ担当者は、次の手順で、Adaptive Server のサーバ証明書とプライベート・キーをインストールします。

- ユーザ環境に導入されている既存のパブリック・キー・インフラストラクチャ (PKI) に用意されているサードパーティのツールを使用する。
- 信頼された第三者の認証局と Adaptive Server 証明書要求ツールを組み合わせて使用する。

証明書を取得するには、CA に証明書を要求してください。第三者に要求した証明書が PKCS #12 フォーマットである場合は、certpk12 ユーティリティを使用して、その証明書を Adaptive Server で認識できるフォーマットに変換してください。

Adaptive Server 証明書要求ツールをテストし、その認証方法がサーバで機能していることを確認するために、Adaptive Server では、ユーザが認証局として機能し、認証局が署名した証明書をユーザ自身に発行できるようにするツールをテスト用に用意しています。

Adaptive Server で使用する証明書を作成するには、次の手順に従います。

1 パブリック・キーとプライベート・キーのペアを生成します。

2 プライベート・キーを安全な場所に保管します。

3 証明書要求を生成します。

4 証明書要求を CA に送信します。

5 認証局が署名した証明書が返されたら、その証明書をファイルに保存し、プライベート・キーを証明書に追加します。

6 Adaptive Server インストール・ディレクトリに証明書を格納します。

証明書を要求するサードパーティ・ツール

ほとんどのサードパーティ PKI ベンダといくつかのブラウザには、証明書とプライベート・キーを生成するユーティリティがあります。これらのユーティリティの多くはグラフィカルなウィザード形式で、一連の質問にユーザが答えると証明書の識別名と共通名が定義されます。

ウィザードの指示に従って、証明書要求を作成します。署名済みの PKCS #12 フォーマット証明書を受け取ったら、certpk12 を使用して、証明書ファイルとプライベート・キー・ファイルを生成します。この 2 つのファイルを連結して *servername.crt* ファイルを作成します。*servername* はサーバ名です。このファイルは、$SYBASE/$SYBASE_ASE の下の *certificates* ディレクトリに置いてください。詳細については、『ユーティリティ・ガイド』を参照してください。

Adaptive Server ツールを使用した証明書の要求と認可

Adaptive Server には、証明書の要求と認証を行う 2 つのツールがあります。certreq は、パブリック・キーとプライベート・キーのペアと証明書要求を生成します。certauth は、サーバ証明書要求を認証局署名済み証明書に変換します。

警告！ certauth は、テスト専用で使用します。商用 CA のサービスを利用することをおすすめします。こうしたサービスではルート証明書の整合性が保護されており、広く承認された CA により署名された証明書を使用すれば、クライアント証明書を使用する形式の認証への移行が促進されるからです。

サーバの信頼されたルート証明書を用意するには、5 つの手順を実行します。最初の 2 つの手順では、テスト版の信頼されたルート証明書を作成します。ここで、サーバ証明書を作成できることを確認できます。検査用の CA 証明書 (信頼されたルート証明書) を作成したら、3 ～ 5 の手順を繰り返してサーバ証明書に署名します。

1　certreq を使用して、証明書を要求します。

2　certauth を使用して、証明書要求を認証局自己署名証明書 (信頼されたルート証明書) に変換します。

3　certreq を使用して、サーバ証明書とプライベート・キーを要求します。

4　certauth を使用して、証明書要求を認証局署名済みサーバ証明書に変換します。

5　プライベート・キーのテキストをサーバ証明書に付加して、サーバのインストール・ディレクトリに証明書を格納します。

注意 Adaptive Server では openssl オープン・ソース・ユーティリティが *$SYBASE/$SYBASE_OCS/bin* に含まれています。certreq、certauth、certpk12 で実装されたすべての証明書管理タスクを実行するには openssl を使用します。Sybase では便宜上このバイナリを組み込んでいますが、バイナリを使用して発生した問題についてはいっさい責任を負いません。詳細については、www.openssl.org を参照してください。

第三者証明書の要求、認証、変換に使用する Sybase ユーティリティ certauth、certreq、certpk12 の詳細については、『ユーティリティ・ガイド』を参照してください。

注意 certauth と certreq は、RSA と DSA のアルゴリズムに依存しています。これらのツールは、RSA アルゴリズムおよび DSA アルゴリズムを使用して証明書要求を構築する暗号モジュールと組み合わせる場合にのみ機能します。

サーバ・ディレクトリ・エントリの作成

Adaptive Server は、クライアント・ログインとサーバ間の RPC を受け入れます。Adaptive Server が接続を受け入れるアドレスやポート番号は設定可能であり、複数のネットワーク、さまざまなプロトコル、代替ポートを指定することができます。

interfaces ファイルでは、SSL は master 行と query 行でのフィルタとして指定しますが、DCE や Kerberos などのセキュリティ・メカニズムを指定するには SECMECH 行を使用します。以下の例は、UNIX 環境で Adaptive Server に SSL を使用する場合の TLI ベース・エントリを示しています。

UNIX で SSL および DCE セキュリティ・メカニズムを使用する Adaptive Server のエントリは、次のように設定します。

```
SYBSRV1
master tcp ether myhostname myport1 ssl
query    tcp ether myhostname myport1 ssl
master tcp ether myhostname myport2
SECMECH 1.3.6.1.4.897.4.6.1
```

Windows で SSL および Kerberos セキュリティ・メカニズムを使用するサーバのエントリは、次のように設定します。

```
[SYBSRV2]
    query=nlwnsck, 18.52.86.120,2748,ssl
    master=nlwnsck 18.52.86.120,2748,ssl
    master=nlwnsck 18.52.86.120,2749
    secmech=1.3.6.1.4.897.4.6.6
```

例における SYBSRV1 と SYBSRV2 の SECMECH 行には、DCE と Kerberos のセキュリティ・メカニズムをそれぞれ参照する OID (オブジェクト識別子) が含まれています。OID の値は次のファイルに定義されています。

- UNIX － *$SYBASE/$SYBASE_OCS/config/objectid.dat*
- Windows － *%SYBASE%¥%SYBASE_OCS¥ini¥objectid.dat*

これらの例では、SSL セキュリティ・サービスはポート番号 2748 (0x0abc) に設定されています。

注意 SSL を SECMECH セキュリティ・メカニズムと同時に使用する意図は、SECMECH から SSL セキュリティへのマイグレーションを容易にすることにあります。

証明書の管理

Adaptive Server で SSL や証明書を管理するには、sp_ssladmin を使用します。このストアド・プロシージャを実行するには sso_role が必要です。

sp_ssladmin では次のことを実行できます。

- ローカル・サーバ証明書を追加する。証明書を追加して、プライベート・キーの暗号化に使用するパスワードを指定することも、起動時にコマンド・ラインからのパスワード入力を要求するようにすることもできる。
- ローカル・サーバ証明書を削除する。
- サーバ証明書のリストを表示する。

sp_ssladmin の構文は次のとおりです。

```
sp_ssladmin {[addcert, certificate_path [, password|NULL]]
      [dropcert, certificate_path]
      [lscert]
      [help]}
      [lsciphers]
      [setciphers, {"FIPS" | "Strong" | "Weak" | "All"
      | quoted_list_of_ciphersuites}]
```

例：

```
sp_ssladmin addcert, "/sybase/ASE-12_5/certificates/Server1.crt",
      "mypassword"
```

この設定により、ローカル・サーバの証明書ファイル *Server1.crt* を、絶対パス */sybase/ASE-12_5/certificates* (Windows の場合は *x:¥sybase¥ASE-12_5¥certificates*) に追加します。プライベート・キーは、パスワード *mypassword*・を使用して暗号化されています。プライベート・キーの作成時に指定したパスワードを指定してください。

証明書を受け入れる前に、sp_ssladmin は次のことを確認します。

- 指定されたパスワードを使用してプライベート・キーを復号化できる (NULL が指定された場合を除く)。
- 証明書のプライベート・キーとパブリック・キーが一致する。
- ルート認証局からサーバ証明書までの証明書チェーンが正しい。
- 証明書内の共通名が、*interfaces* ファイル内の共通名と一致する。

共通名が一致しない場合は、sp_ssladmin は警告を発行します。その他の基準が満たされない場合は、その証明書は証明書ファイルに追加されません。

警告！ Adaptive Server では、パスワードは最大 64 文字です。さらに、プラットフォームによっては、サーバ証明書の作成時に有効なパスワード長が制限されます。次の制限の範囲内でパスワードを選択してください。

- Sun Solaris － 32 ビットおよび 64 ビットの両方のプラットフォーム、最大 256 文字
- Linux － 128 文字
- IBM － 32 ビットおよび 64 ビットの両方のプラットフォーム、32 文字
- HP － 32 ビットおよび 64 ビットの両方のプラットフォーム、8 文字
- Windows － 256 文字

NULL をパスワードとして使用する意図は、SSL 暗号化セッションを開始する前の、SSL の初期設定の間パスワードを保護することにあります。SSL はまだ設定されていないので、パスワードは暗号化されずに接続を介して送られます。最初のログイン時にパスワードを NULL に指定すると、これを防止できます。

NULL をパスワードにした場合は、-y フラグを付けて dataserver を開始する必要があります。このとき、コマンド・ラインでプライベート・キーのパスワードを入力するためのプロンプトが表示されます。

SSL 接続が確立された状態で Adaptive Server を再起動した後、実際のパスワードを使用して sp_ssladmin を再実行します。このパスワードは暗号化されて保管されます。その後、コマンド・ラインから Adaptive Server を起動するときは、この暗号化されたパスワードが使用されるので、管理者が起動時にコマンド・ラインからパスワードを指定する必要がなくなります。

最初のログイン時に NULL のパスワードを使用する方法の代わりに、isql を使用した Adaptive Server へのリモート接続をできないようにするという方法があります。*interfaces* ファイル (Windows では *sql.ini*) 内の *hostname* として “localhost” を指定すると、クライアントはリモート接続できなくなります。ローカル接続だけが確立できるので、パスワードがネットワーク接続を介して転送されることはありません。

注意 Adaptive Server のネットワーク・メモリ・プールには十分なメモリがあるため、sp_ssladmin addcert はデフォルトのメモリ割り付けを使用して証明書とプライベート・キーを設定できます。ただし、ネットワーク・メモリを消費する別のプログラムがデフォルト・ネットワーク・メモリの割り付けを既に行っていた場合、sp_ssladmin は失敗し、次のエラーがクライアントに対して出力されます。

```
Msg 12823, Level 16, State 1:
Server 'servername', Procedure 'sp_ssladmin', Line 72:
```

```
Command 'addcert' failed to add certificate path
/work/REL125/ASE-12_5/certificates/servername.crt, system
error:ErrMemory.
(return status = 1)
```

または、次のメッセージがログ・ファイルに書き込まれます。

```
 ... ssl_alloc:Cannot allocate using ubfalloc(rnetmempool,
131072)
```

対処方法として、additional network memory 設定パラメータの値を大きくすることができます。sp_ssladmin addcert が正常に終了するには約 500K バイトのメモリが必要なので、additional network memory をこの値まで大きくすることで、操作を成功させることができます。このメモリは、必要に応じてネットワーク・メモリ・プールで再使用されます。または、sp_ssladmin が正常に完了した後、additional network memory の値を元に戻すこともできます。

パフォーマンス

安全なセッションの確立に必要な、追加のオーバヘッドがあります。データを暗号化するとサイズが増え、情報の暗号化と復号化に追加の計算が必要になるからです。SSL の追加メモリ要件は、ネットワーク・スループットまたは接続を確立するためのオーバヘッドを 50 ～ 60 パーセント増加させます。ユーザ接続ごとに約 40K のメモリがさらに必要になります。

暗号スイート

SSL ハンドシェイク中に、クライアントとサーバは CipherSuite を介して共通のセキュリティ・プロトコルをネゴシエートします。暗号スイートは、SSL 対応のアプリケーションで使用されるキー交換アルゴリズム、ハッシュ方式、暗号化方式の優先順位付きリストです。暗号スイートの詳細については、IETF (Internet Engineering Task Force) の Web ページ (http://www.ietf.org/rfc/rfc2246.txte) を参照してください。

デフォルトでは、クライアントとサーバの両方がサポートしている最強の CipherSuite が SSL ベースのセッションに使用されます。

Adaptive Server は、SSL Plus ライブラリ API と暗号エンジンである Security Builder™ (両方とも Certicom 製) で使用可能な暗号スイートをサポートしています。

注意　上記にリストした暗号スイートは、TLS (トランスポート・レイヤ仕様) に準拠しています。TLS は SSL 3.0 を拡張したものであり、SSL バージョン 3.0 暗号スイートの別名です。

@@ssl_ciphersuite

Transact-SQL グローバル変数 *@@ssl_ciphersuite* によって、ユーザは SSL ハンドシェイクでどの暗号スイートが選択されたか、また、SSL または非 SSL 接続が確立されているか知ることができます。

Adaptive Server は、SSL ハンドシェイクが完了したときに *@@ssl_ciphersuite* を設定します。値は、非 SSL 接続であることを示す NULL、または SSL ハンドシェイクで選択された暗号スイートの名前を含む文字列のいずれかになります。

たとえば、SSL プロトコルを使用する isql 接続では、この接続で選択された暗号スイートが表示されます。

```
1>  select @@ssl_ciphersuite
2>   go
```

出力：

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

(1 row affected)
```

SSL 暗号スイートの優先度の設定

Adaptive Server の sp_ssladmin には、暗号スイートの優先度を表示および設定するためのコマンド・オプションとして、lsciphers と setciphers という 2 つのコマンド・オプションがあります。これらのオプションによって Adaptive Server が使用する暗号スイートのセットを制限することで、システム・セキュリティ担当者はサーバに対するクライアント接続や Adaptive Server からのアウトバウンド接続で使われる暗号化アルゴリズムの種類をコントロールすることができます。Adaptive Server で SSL 暗号スイートを使用する場合のデフォルトの動作は以前のバージョンと変わりません。暗号スイートのために内部的に定義された優先度セットが使われます。

暗号スイートの優先度セットの値を表示するには、次のように入力します。

```
sp_ssladmin lsciphers
```

特定の暗号スイートの優先度を設定するには次のように入力します。

```
sp_ssladmin setciphers, {"FIPS" | "Strong" | "Weak" | "All" |
quoted_list_of_ciphersuites }
```

各パラメータの意味は、次のとおりです。

- "FIPS" － 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に含まれない暗号化アルゴリズムのセット。

- “All” − デフォルトの暗号スイートのセット。
- quoted_list_of_ciphersuites − 暗号スイートのセットを、優先度順にカンマで区切ったリストで指定します。引用符 (“) でリストの先頭と最後をマークします。引用符で囲んだリストに、個々の暗号スイート名のほか、定義済みの任意のセットを含めることができます。未知の暗号スイート名を指定するとエラーが報告され、優先度は変更されません。

定義済みのセットの詳細な内容については、表 19-1 (668 ページ) を参照してください。

sp_ssladmin setciphers は、指定された順序リストに暗号スイートの優先度を設定します。これは使用可能な SSL 暗号スイートを、“FIPS”、“Strong”、“Weak”、“All”、または引用符で囲まれた暗号スイート・リストのセットに制限します。これが有効になるのは次のリスナが開始されたときで、Adaptive Server を再起動してすべてのリスナが新しい設定を使うようにする必要があります。

設定されている任意の暗号スイートの優先度を、sp_ssladmin lsciphers で表示することができます。優先度が設定されていない場合、sp_ssladmin lsciphers は 0 個のローを返します。これは優先度が設定されておらず、Adaptive Server がデフォルトの (内部) 優先度を使うことを意味します。

表 *19-1: Adaptive Server* の定義済み暗号スイート

セット名	セット内の暗号スイート名
FIPS	TLS_RSA_WITH_AES_256_CBC_SHA TLS_RSA_WITH_AES_128_CBC_SHA TLS_RSA_WITH_3DES_EDE_CBC_SHA TLS_DHE_DSS_WITH_3DES_EDE_CBC_SHA TLS_DHE_RSA_WITH_3DES_EDE_CBC_SHA TLS_RSA_WITH_DES_CBC_SHA TLS_DHE_DSS_WITH_DES_CBC_SHA TLS_DHE_RSA_WITH_DES_CBC_SHA TLS_RSA_EXPORT1024_WITH_DES_CBC_SHA TLS_DHE_DSS_EXPORT1024_WITH_DES_CBC_SHA
Strong	TLS_RSA_WITH_AES_256_CBC_SHA TLS_RSA_WITH_AES_128_CBC_SHA TLS_RSA_WITH_3DES_EDE_CBC_SHA TLS_RSA_WITH_RC4_128_SHA TLS_RSA_WITH_RC4_128_MD5 TLS_DHE_DSS_WITH_3DES_EDE_CBC_SHA TLS_DHE_DSS_WITH_RC4_128_SHA TLS_DHE_RSA_WITH_3DES_EDE_CBC_SHA
Weak	TLS_RSA_WITH_DES_CBC_SHA TLS_DHE_DSS_WITH_DES_CBC_SHA TLS_DHE_RSA_WITH_DES_CBC_SHA TLS_RSA_EXPORT1024_WITH_DES_CBC_SHA TLS_RSA_EXPORT1024_WITH_RC4_56_SHA TLS_DHE_DSS_EXPORT1024_WITH_RC4_56_SHA TLS_DHE_DSS_EXPORT1024_WITH_DES_CBC_SHA TLS_RSA_EXPORT_WITH_RC4_40_MD5 TLS_RSA_EXPORT_WITH_DES40_CBC_SHA TLS_DHE_DSS_EXPORT_WITH_DES40_CBC_SHA TLS_DHE_RSA_EXPORT_WITH_DES40_CBC_SHA

セット名	セット内の暗号スイート名
すべて	TLS_RSA_WITH_AES_256_CBC_SHA
	TLS_RSA_WITH_AES_128_CBC_SHA
	TLS_RSA_WITH_3DES_EDE_CBC_SHA
	TLS_RSA_WITH_RC4_128_SHA
	TLS_RSA_WITH_RC4_128_MD5
	TLS_DHE_DSS_WITH_3DES_EDE_CBC_SHA
	TLS_DHE_DSS_WITH_RC4_128_SHA
	TLS_DHE_RSA_WITH_3DES_EDE_CBC_SHA
	TLS_RSA_WITH_DES_CBC_SHA
	TLS_DHE_DSS_WITH_DES_CBC_SHA
	TLS_DHE_RSA_WITH_DES_CBC_SHA
	TLS_RSA_EXPORT1024_WITH_DES_CBC_SHA
	TLS_RSA_EXPORT1024_WITH_RC4_56_SHA
	TLS_DHE_DSS_EXPORT1024_WITH_RC4_56_SHA
	TLS_DHE_DSS_EXPORT1024_WITH_DES_CBC_SHA TLS_RSA_EXPORT_WITH_RC4_40_MD5
	TLS_RSA_EXPORT_WITH_DES40_CBC_SHA
	TLS_DHE_DSS_EXPORT_WITH_DES40_CBC_SHA
	TLS_DHE_RSA_EXPORT_WITH_DES40_CBC_SHA

表 19-2 は、Adaptive Server 15.0 以降ではサポートされない暗号スイートを示します。15.0. 削除された暗号スイートを使用すると SSLHandshake が失敗し、Adaptive Server には接続できません。

表 *19-2:* 削除された暗号スイート

セット名	セットから削除された暗号スイート名
FIPS	TLS_RSA_EXPORT1024_WITH_DES_CBC_SHA
Strong	削除なし
Weak	TLS_RSA_EXPORT1024_WITH_RC4_56_SHA
	TLS_DHE_DSS_EXPORT_WITH_DES40_CBC_SHA
	TLS_DHE_RSA_EXPORT_WITH_DES40_CBC_SHA
その他のデッドロック	TLS_DH_anon_EXPORT_WITH_DES40_CBC_SHA
	TLS_DH_anon_EXPORT_WITH_RC4_40_MD5
	TLS_DH_anon_WITH_3DES_EDE_CBC_SHA
	TLS_DH_anon_WITH_DES_CBC_SHA
	TLS_DH_anon_WITH_RC4_128_MD5
	TLS_RSA_WITH_NULL_MD5
	TLS_RSA_WITH_NULL_SHA

sp_ssladmin の例

最初に開始されるときは、まだ暗号スイートの優先度が設定されていないので、sp_ssladmin lscipher は優先度を表示しません。

```
1> sp_ssladmin lsciphers
2> go
```

出力：

```
 Cipher Suite Name   Preference
-----------------   ----------
(0 rows affected)
(return status = 0)
```

次の例では、FIPS アルゴリズムを使用する暗号スイートのセットを指定しています。

```
11> sp_ssladmin setcipher, 'FIPS'
The following cipher suites and order of preference are set for SSL connections:

Cipher Suite Name                                                Preference
---------------------------------------------------------------- -----------
TLS_RSA_WITH_AES_256_CBC_SHA                                               1
TLS_RSA_WITH_AES_128_CBC_SHA                                               2
TLS_RSA_WITH_3DES_EDE_CBC_SHA                                              3
TLS_DHE_DSS_WITH_3DES_EDE_CBC_SHA                                          4
TLS_DHE_RSA_WITH_3DES_EDE_CBC_SHA                                          5
TLS_RSA_WITH_DES_CBC_SHA                                                   6
TLS_DHE_DSS_WITH_DES_CBC_SHA                                               7
TLS_DHE_RSA_WITH_DES_CBC_SHA                                               8
TLS_RSA_EXPORT1024_WITH_DES_CBC_SHA                                        9
TLS_DHE_DSS_EXPORT1024_WITH_DES_CBC_SHA                                     10
```

優先度 0 (ゼロ) の sp_ssladmin 出力は、Adaptive Server で使用されない暗号スイートを示します。他のゼロ以外の値は、SSL ハンドシェイクの間に Adaptive Server がアルゴリズムを使用する優先度の順序を示します。SSL ハンドシェイクのクライアント側はこれらの暗号スイートから、受け付ける暗号スイートのリストに一致するものを選びます。

この例では、引用符で囲んだ暗号スイートのリストで、Adaptive Server に優先度を設定しています。

```
1> sp_ssladmin setcipher, 'TLS_RSA_WITH_AES_128_CBC_SHA,
TLS_RSA_WITH_AES_256_CBC_SHA'
2> go
The following cipher suites and order of preference are set for SSL connections:
Cipher Suite Name                                                Preference
---------------------------------------------------------------- -----------
TLS_RSA_WITH_AES_128_CBC_SHA                                               1
TLS_RSA_WITH_AES_256_CBC_SHA                                               2
```

その他の注意事項

Adaptive Server バージョン 12.5.3 以降にアップグレードしたときは、サーバのデフォルトが暗号スイートの優先度になり、sp_ssladmin のオプション lscipher で優先度は表示されません。サーバはそのデフォルトの優先度、すなわち “All” で定義される優先度を使用します。システム・セキュリティ担当者は、自分のサイトのセキュリティ・ポリシーと使用可能な SSL 暗号スイートを検討し、暗号スイートを制限するかどうかや、どの暗号スイートがセキュリティ・ポリシーに合っているかを判断する必要があります。

Adaptive Server バージョン 12.5.3 以降からアップグレードするときに、暗号スイートの優先度が設定されている場合、設定された優先度がアップグレード後も使用されます。アップグレードの完了後に、サーバの暗号スイートの優先度が現在のセキュリティ・ポリシーに適合していることを確認し、表 19-1 の暗号スイート・リストで暗号スイートがサポートされているかどうかを調べてください。サポートされていない暗号スイートは削除してください。

設定した SSL 暗号スイートの優先度をサーバからすべて削除してデフォルトの優先度を使いたい場合は、次のコマンドを使用してシステム・カタログ内の記憶領域から優先度を削除します。

```
1> sp_configure 'allow updates to system tables', 1
2> go

1> delete from master..sysattributes where class=24
2> go

1> sp_configure 'allow updates to system tables', 0
2> go
```

これらのコマンドは、システム・セキュリティ担当者またはシステム管理者だけが実行できます。

SSL を使用した共通名の指定

ディレクトリ・サービス・エントリで指定したサーバ名は、SSL ハンドシェイクを実行する際に SSL サーバ証明書が使用する共通名とは異なる可能性があります。これにより、SSL 証明書の共通名の完全修飾ドメイン名 (たとえば、*server1.bigcompany.com*) を使用できます。

interfaces ファイルに共通名を追加するには、次のコマンドを使用します。

```
ase1
   master tcp ether host_name port_number ssl="CN='common_name'"
   query tcp ether host_name port_number ssl="CN='common_name'"
```

SSL を使用する Adaptive Server にクライアントが SSL を使用して接続する場合は、*interfaces* ファイルのポート番号の後に SSL フィルタが配置されます。ディレクトリ・サービスには、dsedit またはテキスト・エディタを使用して追加できる共通名が含まれます。

sp_listener での共通名の指定

sp_listener にはパラメータ CN=*common_name* が含まれており、SSL 証明書の共通名を指定できます。構文は次のとおりです。

```
sp_listener 'command','[protocol:]machine_name:port_number:
"CN=common_name"', 'engine_number'
```

プロトコルとして ssltcp を指定する場合にのみ、CN=*common_name* を使用します。ここで指定する *common_name* は SSL 証明書の *common_name* に照らして検証されます。CN=*common_name* を含めない場合、Adaptive Server は *server_name* を使用して SSL 証明書の共通名に照らして検証します。証明書に完全修飾ドメイン名を含める場合、このドメイン名は CN=*common_name* と一致する必要があります。

属性名 “CN” は大文字と小文字を区別しません (“CN”、“cn”、または “Cn” を使用できます) が、共通名の属性値は大文字と小文字を区別します。

たとえば、共通名 `ase1.big server 1.com` を指定するには、次のように入力します。

```
sp_listener 'start','ssltcp:blade1:17251:"CN=ase1.big server 1.com"','0'
```

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

変更されたストアド・プロシージャ *sp_addserver*

filter パラメータは、共通名を指定するように拡張されています。『リファレンス・マニュアル：プロシージャ』を参照してください。

Kerberos による機密保持

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

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

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

Kerberos やサポートされているその他のセキュリティ・サービスを使用したメッセージの機密保持の詳細については、「ネットワークベース・セキュリティの管理」(473 ページ) を参照してください。

パスワード保護を使用したデータベースのダンプとロード

dump database コマンドの password パラメータを使用すると、権限を持たないユーザがデータベース・ダンプをロードできないように保護することができます。データベース・ダンプの作成時に password パラメータを指定した場合には、データベースのロード時にもこのパスワードを指定する必要があります。

パスワード保護に対応する dump database コマンドと load database コマンドの構文の一部は次のとおりです。

dump database *database_name* to *file_name* [with passwd = *password*]
load database *database_name* from *file_name* [with passwd = *password*]

各パラメータの意味は、次のとおりです。

- *database_name* − ダンプまたはロードするデータベースの名前です。
- *file_name* − ダンプ・ファイルの名前です。
- *password* − 不正なユーザからダンプ・ファイルを保護するために指定するパスワードです。

6 文字より短く 30 文字より長いパスワードを指定すると、Adaptive Server からエラー・メッセージが発行されます。データベースをロードするときに誤ったパスワードを発行すると、Adaptive Server からエラー・メッセージが発行され、コマンドは失敗します。

たとえば、次の例はパスワード “bluesky” を使用して pubs2 データベースのデータベース・ダンプを保護します。

```
dump database pubs2 to ・Syb_backup/mydb.db・with passwd = “bluesky”
```

このデータベース・ダンプをロードするときには同じパスワードを使用する必要があります。

```
load database pubs2 from ・Syb_backup/mydb.db・with passwd = “bluesky”
```

パスワードと以前のバージョンの Adaptive Server

パスワード保護に対応する dump コマンドと load コマンドを使用できるのは、Adaptive Server バージョン 12.5.2 以降のみです。Adaptive Server バージョン 12.5.2 のダンプに password パラメータを使用した場合、そのダンプを以前のバージョンの Adaptive Server にロードしようとすると失敗します。

パスワードと文字セット

ダンプをロードできるサーバは同じ文字セットを使用しているサーバのみです。たとえば、ASCII 文字セットを使用するサーバから ASCII 以外の文字セットを使用するサーバにダンプをロードしようとすると、ASCII のパスワードの値は ASCII ではないパスワードと異なるためロードが失敗します。

ユーザが入力したパスワードは、Adaptive Server のローカル文字セットに変換されます。ASCII 文字は通常は文字セット間で値の表現が同じであるため、ユーザのパスワードが ASCII 文字セットであれば、dump と load のパスワードはすべての文字セットで認識されます。

Adaptive Server バージョン 15.0.2 以降では、ポータブル・パスワードを保存できます。「パスワード文字セットの考慮事項」(443 ページ) を参照してください。

索引

記号

数字

A

B

C

D

E

F

G

H

I

J

K

L

M

N

O

P

Q

R

S

T

U

V

W

X

あ

い

う

え

お

か

き

く

け

こ

さ

し

す

せ

そ

た

ち

つ

て

と

な

に

ね

の

は

ひ

ふ

へ

ほ

ま

め

も

や

ゆ

よ

ら

り

る

れ

ろ

わ