iAnywhere SOLUTIONS™
A SYBASE COMPANY

SQL Anywhere® Studio
セキュリティ・ガイド

パート番号：DC03973-01-0902-01
改訂：2005 年 3 月

目次

はじめに

このマニュアルの内容

このマニュアルでは、SQL Anywhere Studio で使用可能なセキュリティ機能について説明します。基本的なセキュリティ情報に加えて、C2 基準を満たした環境と互換性のある形で、現在のバージョンの SQL Anywhere Studio を運用する方法について説明します。

このマニュアルには、セキュリティ関連の機能に関するすべての情報が含まれているわけではありません。

現在のソフトウェアは C2 公認ではない

Adaptive Server Anywhere バージョン 7.0.0 は、米国連邦政府の C2 セキュリティ証明を取得しました。このマニュアルの「C2」の項では、C2 基準を満たした構成と互換性のある形で、Adaptive Server Anywhere の現在のバージョンを運用する方法について説明します。

このマニュアルは、C2 準拠について説明した公認マニュアルでは*ありません*。公認マニュアル (英語版) は、Sybase の Web サイト http://www.sybase.com/detail?id=1010458 で入手できます。このマニュアルの記述によって、このソフトウェアの現在のバージョンが C2 準拠であるとは解釈しないでください。「C2 基準を満たした構成と同様」およびこれに類似した文章が使用されますが、これは実際の C2 準拠を意味するものではありません。C2 基準を満たした方法で運用する*唯*一の方法は、C2 公認マニュアルに従って C2 基準を満たしたリリースのソフトウェアを使用することです。

対象読者

このマニュアルは、Adaptive Server Anywhere のセキュリティ機能を利用するユーザ、または C2 基準を満たした構成と同じ方法で Adaptive Server Anywhere を実行するユーザを対象としています。

SQL Anywhere Studio のマニュアル

このマニュアルは、SQL Anywhere のマニュアル・セットの一部です。この項では、マニュアル・セットに含まれる各マニュアルと使用法について説明します。

SQL Anywhere Studio のマニュアル

SQL Anywhere Studio のマニュアルは、各マニュアルを 1 つの大きなヘルプ・ファイルにまとめたオンライン形式、マニュアル別の PDF ファイル、および有料の製本版マニュアルで提供されます。SQL Anywhere Studio のマニュアルは、次の分冊マニュアルで構成されています。

- **『SQL Anywhere Studio の紹介』** このマニュアルでは、SQL Anywhere Studio のデータベース管理と同期テクノロジの概要について説明します。また、SQL Anywhere Studio を構成する各部分について説明するチュートリアルも含まれています。
- **『SQL Anywhere Studio 新機能ガイド』** このマニュアルは、SQL Anywhere Studio のこれまでのリリースのユーザを対象としています。ここでは、製品の今回のリリースと以前のリリースで導入された新機能をリストし、アップグレード手順を説明しています。
- **『Adaptive Server Anywhere データベース管理ガイド』** このマニュアルでは、データベースおよびデータベース・サーバの実行、管理、設定について説明しています。
- **『Adaptive Server Anywhere SQL ユーザーズ・ガイド』** このマニュアルでは、データベースの設計と作成の方法、データのインポート・エクスポート・変更の方法、データの検索方法、ストアド・プロシージャとトリガの構築方法について説明します。
- **『Adaptive Server Anywhere SQL リファレンス・マニュアル』** このマニュアルは、Adaptive Server Anywhere で使用する SQL 言語の完全なリファレンスです。また、Adaptive Server Anywhere のシステム・テーブルとシステム・プロシージャについても説明しています。
- **『Adaptive Server Anywhere プログラミング・ガイド』** このマニュアルでは、C、C++、Java プログラミング言語を使用してデータベース・アプリケーションを構築、配備する方法につい

て説明します。Visual Basic や PowerBuilder などのツールのユーザは、それらのツールのプログラミング・インタフェースを使用できます。また、Adaptive Server Anywhere ADO.NET データ・プロバイダについても説明します。

- **『Adaptive Server Anywhere SNMP Extension Agent ユーザーズ・ガイド』** このマニュアルでは、Adaptive Server Anywhere SNMP Extension Agent を SNMP 管理アプリケーションとともに使用できるように設定して、Adaptive Server Anywhere データベースを管理できるようにする方法を説明します。

- **『Adaptive Server Anywhere エラー・メッセージ』** このマニュアルでは、Adaptive Server Anywhere エラー・メッセージの完全なリストを、その診断情報とともに説明します。

- **『SQL Anywhere Studio セキュリティ・ガイド』** このマニュアルでは、Adaptive Server Anywhere データベースのセキュリティ機能について説明します。Adaptive Server Anywhere 7.0 は、米国政府から TCSEC (Trusted Computer System Evaluation Criteria) の C2 セキュリティ評価を授与されています。このマニュアルには、Adaptive Server Anywhere の現在のバージョンを、C2 基準を満たした環境と同等の方法で実行することを望んでいるユーザにとって役に立つ情報が含まれています。

- **『Mobile Link 管理ガイド』** このマニュアルでは、モバイル・コンピューティング用の Mobile Link データ同期システムについてあらゆる角度から説明します。このシステムによって、Oracle、Sybase、Microsoft、IBM の単一データベースと、Adaptive Server Anywhere や Ultra Light の複数データベースの間でのデータ共有が可能になります。

- **『Mobile Link クライアント』** このマニュアルでは、Adaptive Server Anywhere リモート・データベースと Ultra Light リモート・データベースの設定を行い、これらを同期させる方法について説明します。

- **『Mobile Link サーバ起動同期ユーザーズ・ガイド』** このマニュアルでは、Mobile Link のサーバによって開始される同期について説明します。サーバによって開始される同期とは、統合データベースから同期の開始を可能にする Mobile Link の機能です。

- **『Mobile Link チュートリアル』** このマニュアルには、Mobile Link アプリケーションの設定と実行を行う方法を説明するチュートリアルがいくつか用意されています。
- **『QAnywhere ユーザーズ・ガイド』** このマニュアルでは、Mobile Link QAnywhere について説明します。Mobile Link QAnywhere は、従来のデスクトップ・クライアントやラップトップ・クライアントだけでなく、モバイル・クライアントや無線クライアント用のメッセージング・アプリケーションの開発と展開を可能にするメッセージング・プラットフォームです。
- **『Mobile Link およびリモート・データ・アクセスの ODBC ドライバ』** このマニュアルでは、Mobile Link 同期サーバから、または Adaptive Server Anywhere リモート・データ・アクセスによって、Adaptive Server Anywhere 以外の統合データベースにアクセスするための ODBC ドライバの設定方法について説明します。
- **『SQL Remote ユーザーズ・ガイド』** このマニュアルでは、モバイル・コンピューティング用の SQL Remote データ・レプリケーション・システムについて、あらゆる角度から説明します。このシステムによって、Adaptive Server Anywhere または Adaptive Server Enterprise の単一データベースと Adaptive Server Anywhere の複数データベースの間で、電子メールやファイル転送などの間接的リンクを使用したデータ共有が可能になります。
- **『SQL Anywhere Studio ヘルプ』** このマニュアルには、Sybase Central や Interactive SQL、その他のグラフィカル・ツールに関するコンテキスト別のヘルプが含まれています。これは、製本版マニュアル・セットには含まれていません。
- **『Ultra Light データベース・ユーザーズ・ガイド』** このマニュアルは、Ultra Light 開発者を対象としています。ここでは、Ultra Light データベース・システムの概要について説明します。また、すべての Ultra Light プログラミング・インタフェースに共通する情報を提供します。
- **Ultra Light のインタフェースに関するマニュアル** 各 Ultra Light プログラミング・インタフェースには、それぞれに対応するマニュアルを用意しています。これらのインタフェースは、RAD (

ラピッド・アプリケーション開発) 用の Ultra Light コンポーネントとして提供されているものと、C、C++、Java 開発用の静的インタフェースとして提供されているものがあります。

このマニュアル・セットの他に、PowerDesigner と InfoMaker には、独自のオンライン・マニュアル (英語版) がそれぞれ用意されています。

マニュアルの形式

SQL Anywhere Studio のマニュアルは、次の形式で提供されています。

- **オンライン・マニュアル** オンライン・マニュアルには、SQL Anywhere Studio の完全なマニュアルがあり、SQL Anywhere ツールに関する印刷マニュアルとコンテキスト別のヘルプの両方が含まれています。オンライン・マニュアルは、製品のメンテナンス・リリースごとに更新されます。これは、最新の情報を含む最も完全なマニュアルです。

  Windows オペレーティング・システムでオンライン・マニュアルにアクセスするには、[スタート] ー [プログラム] ー [SQL Anywhere 9] ー [オンライン・マニュアル] を選択します。オンライン・マニュアルをナビゲートするには、左ウィンドウ枠で HTML ヘルプの目次、索引、検索機能を使用し、右ウィンドウ枠でリンク情報とメニューを使用します。

  UNIX オペレーティング・システムでオンライン・マニュアルにアクセスするには、SQL Anywhere のインストール・ディレクトリに保存されている HTML マニュアルを参照してください。

- **PDF 版マニュアル** SQL Anywhere の各マニュアルは、Adobe Acrobat Reader で表示できる PDF ファイルで提供されています。

  PDF 版マニュアルは、オンライン・マニュアルまたは Windows の [スタート] メニューから利用できます。

- **製本版マニュアル** 製本版マニュアルをご希望の方は、ご購入いただいた販売代理店または弊社営業担当までご連絡ください。

表記の規則

この項では、このマニュアルで使用されている書体およびグラフィック表現の規則について説明します。

SQL 構文の表記規則

SQL 構文の表記には、次の規則が適用されます。

- **キーワード** SQL キーワードはすべて次の例に示す ALTER TABLE のように大文字で表記します。

  ALTER TABLE [*owner*.]*table-name*

- **プレースホルダ** 適切な識別子または式で置き換えられる項目は、次の例に示す *owner* や *table-name* のように表記します。

  ALTER TABLE [*owner*.]*table-name*

- **繰り返し項目** 繰り返し項目のリストは、次の例に示す *column-constraint* のように、リストの要素の後ろに省略記号 (ピリオド 3 つ …) を付けて表します。

  ADD *column-definition* [*column-constraint, …*]

  複数の要素を指定できます。複数の要素を指定する場合は、各要素間をカンマで区切る必要があります。

- **オプション部分** 文のオプション部分は角カッコで囲みます。

  RELEASE SAVEPOINT [*savepoint-name*]

  この例では、角カッコで囲まれた *savepoint-name* がオプション部分です。角カッコは入力しないでください。

- **オプション** 項目リストから 1 つだけ選択するか、何も選択しなくてもよい場合は、項目間を縦線で区切り、リスト全体を角カッコで囲みます。

  [**ASC** | **DESC**]

  この例では、ASC と DESC のどちらか 1 つを選択しても、どちらも選択しなくてもかまいません。角カッコは入力しないでください。

- **選択肢**　オプションの中の 1 つを必ず選択しなければならない場合は、選択肢を中カッコで囲み、縦棒で区切ります。

  [**QUOTES** { **ON** | **OFF** }]

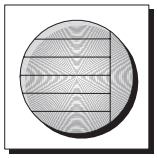
  QUOTES オプションを使用する場合は、ON または OFF のどちらかを選択する必要があります。角カッコと中カッコは入力しないでください。

グラフィック・アイコン

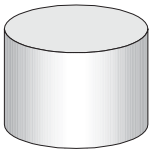
このマニュアルでは、次のアイコンを使用します。

- クライアント・アプリケーション

- Sybase Adaptive Server Anywhere などのデータベース・サーバ



- データベース。高度な図では、データベースとデータベースを管理するデータ・サーバの両方をこのアイコンで表します。



- レプリケーションまたは同期のミドルウェア。ソフトウェアのこれらの部分は、データベース間のデータ共有を支援します。たとえば、Mobile Link 同期サーバ、SQL Remote Message Agent などがあげられます。

- プログラミング・インタフェース

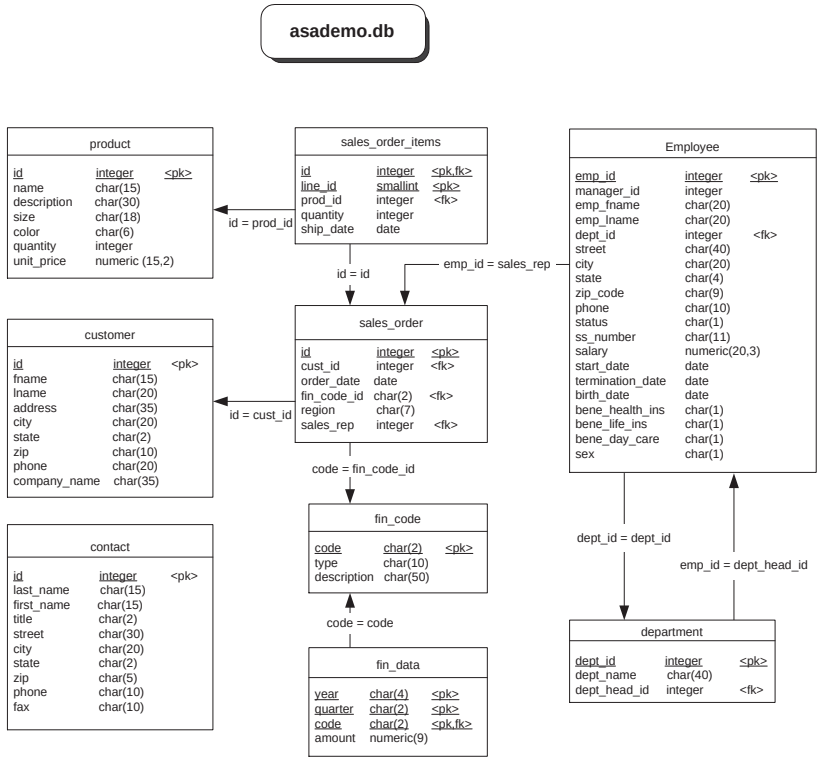
Adaptive Server Anywhere サンプル・データベース

このマニュアルでは、多くの例で Adaptive Server Anywhere サンプル・データベースが使用されています。

サンプル・データベースは、*asademo.db* という名前のファイルに保存され、SQL Anywhere ディレクトリに置かれています。

サンプル・データベースは小規模の企業の例です。データベースには、この企業の内部情報 (従業員、部署、経理) とともに、製品情報や販売情報 (受注、顧客、連絡先) が入っています。データベースに含まれる情報はすべて架空のものです。

次の図は、サンプル・データベース内のテーブルと各テーブル間の関係を示しています。



詳細情報の検索／フィードバックの提供

このマニュアルに関するご意見、ご提案、フィードバックをお寄せください。

マニュアルおよびソフトウェアに関するフィードバックは、SQL Anywhere のテクノロジについて議論するニュースグループを介してお送りいただけます。ニュースグループは、ニュース・サーバ *forums.sybase.com* にあります (ニュースグループにおけるサービスは英語でのみの提供となります)。

以下のニュースグループがあります。

- sybase.public.sqlanywhere.general
- sybase.public.sqlanywhere.linux
- sybase.public.sqlanywhere.mobilink
- sybase.public.sqlanywhere.product_futures_discussion
- sybase.public.sqlanywhere.replication
- sybase.public.sqlanywhere.ultralite
- ianywhere.public.sqlanywhere.qanywhere

ニュースグループに関するお断り

iAnywhere Solutions は、ニュースグループ上に解決策、情報、または意見を提供する義務を負うものではありません。また、システム・オペレータ以外のスタッフにこのサービスを監視させて、操作状況や可用性を保証する義務もありません。

iAnywhere Solutions のテクニカル・アドバイザとその他のスタッフは、時間のある場合にかぎりニュースグループでの支援を行います。こうした支援は基本的にボランティアで行われるため、解決策や情報を定期的に提供できるとはかぎりません。支援できるかどうかは、スタッフの仕事量に左右されます。

マニュアルに関するご意見、ご提案は、SQL Anywhere ドキュメンテーション・チームの iasdoc@ianywhere.com 宛てに電子メールでお寄せください。このアドレスに送信された電子メールに返信する責任は負いませんが、お寄せ頂いたご意見、ご提案は必ず読ませて頂きます。

第 1 部　SQL Anywhere Studio のセキュリティ機能

第 1 部では、SQL Anywhere Studio の基本的なセキュリティ機能について説明します。

第 1 章

安全なデータの管理

この章の内容

この章では、データベースの安全管理に役立つ Adaptive Server Anywhere の機能について説明します。特に、監査、データベースの暗号化、Windows CE データベースの保護について説明します。他のセキュリティ機能の概要についても説明し、詳細情報の参照先も示しています。

データのセキュリティについては、データベース管理者に責任があります。この章で記述されているタスクを実行するには、特に明記されていないかぎり、DBA 権限が必要です。

セキュリティ関連の主なトピックは、ユーザ ID とパーミッションです。これらのトピックについては、『ASA データベース管理ガイド』>「ユーザ ID とパーミッションの管理」を参照してください。

クライアント／サーバ通信の暗号化についての詳細は、「Adaptive Server Anywhere トランスポート・レイヤ・セキュリティ」33 ページを参照してください。

セキュリティ機能の概要

データベースには機密の情報や個人的な情報などが含まれている場合があるので、データベースやそこに含まれるデータのセキュリティを考慮した設計になっていることが重要です。

Adaptive Server Anywhere には、データの安全な環境の構築に役立ついくつかの機能があります。

- **ユーザ ID と認証** データベースにアクセスするユーザを制御します。

  詳細については、『ASA データベース管理ガイド』>「新しいユーザの作成」を参照してください。

- **任意アクセス制御機能** データベースに接続している間にユーザが実行するアクションを制御します。

  詳細については、『ASA データベース管理ガイド』>「データベースのパーミッションの概要」を参照してください。

- **監査** データベースで行われたアクションの記録を管理するのに役立ちます。

  詳細については、「データベース・アクティビティの監査」10 ページを参照してください。

- **データベース・サーバ・オプション** データベースのロードなどの操作を実行するユーザを指定します。このオプションは、データベース・サーバの起動時に設定されます。

  詳細については、『ASA データベース管理ガイド』>「コマンド・ラインからパーミッションを制御する」を参照してください。

- **ビューとストアド・プロシージャ** ユーザがアクセスするデータとユーザが実行する操作を指定します。

  詳細については、『ASA データベース管理ガイド』>「高度なセキュリティを実現するためのビューとプロシージャの使い方」を参照してください。

- **データベースの暗号化**　データベース暗号化機能を使用する際の、暗号化のレベルを選択します。データベースを安全に管理するために、簡単な暗号化または高度な暗号化のいずれかを選択できます。簡単な暗号化は、難読化と同じです。高度な暗号化にすると、暗号化キーなしではデータベースにまったくアクセスできなくなります。

  詳細については、『ASA データベース管理ガイド』>「-ek データベース・オプション」と『ASA データベース管理ガイド』>「DatabaseKey 接続パラメータ [DBKEY]」を参照してください。

- **トランスポート・レイヤ・セキュリティ**　トランスポート・レイヤ・セキュリティを使用すると、クライアント・アプリケーションとデータベース・サーバ間の通信を認証することができます。トランスポート・レイヤ・セキュリティでは、楕円曲線暗号方式または RSA 暗号方式を使用します。

  別途ライセンスを取得できるオプションが必要
  トランスポート・レイヤ・セキュリティを使用するには、別途ライセンスを取得できる SQL Anywhere Studio セキュリティ・オプションを入手する必要があります。このセキュリティ・オプションは、輸出規制対象品目です。

  このコンポーネントのご注文については、『SQL Anywhere Studio の紹介』>「別途ライセンスが入手可能なコンポーネント」を参照してください。

  詳細については、「Adaptive Server Anywhere トランスポート・レイヤ・セキュリティ」33 ページを参照してください。

- **C2 基準**　C2 は米国政府が政府組織内での一貫性を維持するために制定したセキュリティ・ガイドラインです。Adaptive Server Anywhere 7.0 (英語版) を実行しており、対応するハードウェアがある場合は、C2 基準を満たす方法で実行するようにマシンを設定できます。C2 公認マニュアル (英語版) は、http://www.sybase.com/detail?id=1010458 から入手できます。

  Adaptive Server Anywhere の現在のバージョンを C2 基準を満たした構成と同様の環境で実行する方法については、「インストール」57 ページを参照してください。

データベース・アクセスの制御

データベース管理者は、ユーザ ID とパスワードを割り当てることによって、どのユーザがデータベースにアクセスするかを制御します。各ユーザ ID にパーミッションを付与することによって、接続しているときに各ユーザが実行できるタスクを制御します。この項では、データベース・アクセスの管理に使用できる機能について説明します。

ユーザ ID に基づくパーミッションのスキーム

ユーザは、データベースにログインすると、次の基準の*いずれか*を満たすすべてのデータベース・オブジェクトにアクセスできます。

- 自分で作成したオブジェクト
- 明示的なパーミッションが付与されているオブジェクト
- ユーザの所属グループに明示的なパーミッションが付与されているオブジェクト

ユーザは、この基準を満たさないデータベース・オブジェクトにはアクセスできません。つまりユーザは、自分が所有するオブジェクト、またはアクセス権限を明示的に付与されているオブジェクトにのみアクセスできます。

詳細については、次の項目を参照してください。

- 『ASA データベース管理ガイド』> 「ユーザ ID とパーミッションの管理」
- 『ASA SQL リファレンス・マニュアル』> 「CONNECT 文 [ESQL] [Interactive SQL]」
- 『ASA SQL リファレンス・マニュアル』> 「GRANT 文」
- 『ASA SQL リファレンス・マニュアル』> 「REVOKE 文」

統合化ログインの使用方法

統合化ログインを使用すると、ユーザは 1 つのログイン名とパスワードで Windows オペレーティング・システムとデータベースの両方にログインできます。外部ログイン名は、データベース・ユーザ ID に関連付けられています。統合化ログインを行う場合、ユーザはログイン名とパスワードの両方を指定してオペレーティング・システムにログオンします。オペレーティング・システムはそのユーザをサーバに

通知し、サーバは関連付けられたデータベース・ユーザ ID としてそのユーザをログインさせます。追加のログイン名とパスワードは必要ありません。

統合化ログインには、セキュリティに関して検討を必要とする事項があります。たとえば、ユーザ・プロファイル Guest でブランクのパスワードを使用できるようにしておくと、そのサーバで管理しているデータベースへのアクセスが無制限に許可されます。ユーザはデフォルトではログイン時に Guest ユーザのプロファイルを使用するため、現実には、どのユーザでも任意のログイン ID やパスワードを使用してサーバにログインできてしまいます。

詳細については、次の項目を参照してください。

- 『ASA データベース管理ガイド』> 「セキュリティについての考慮事項：無制限データベース・アクセス」
- 『ASA データベース管理ガイド』> 「統合化ログインの使用方法」
- 『ASA データベース管理ガイド』> 「LOGIN_MODE オプション [データベース]」

パスワードのセキュリティの強化

パスワードは、データベースのセキュリティ・システムの重要な部分です。安全のために、パスワードは容易に推測できないものにし、ハード・ドライブやその他のロケーションから簡単にアクセスできないようにしてください。

最小長のパスワードの実装

デフォルトでは、パスワードは任意の長さで指定できます。セキュリティを強化するために、新しいパスワードに必要な最小長を課すことができます。これを実行するには、MIN_PASSWORD_LENGTH データベース・オプションを 0 より大きな値に設定します。次の文は、パスワードが最低でも 8 バイトの長さになるようにします。

```
SET OPTION PUBLIC.MIN_PASSWORD_LENGTH = 8
```

詳細については、『ASA データベース管理ガイド』> 「MIN_PASSWORD_LENGTH オプション [データベース]」を参照してください。

ODBC データ・ソースにパスワードを含めない

パスワードはデータベースへのアクセスのキーとなります。そのため、セキュリティを考慮した環境では、権限のないユーザが簡単にパスワードを使用できないようにすることが重要です。

ODBC データ・ソースを作成するとき、または Sybase Central 接続プロファイルを作成するときに、オプションでパスワードを含めることができます。セキュリティを強化するために、パスワードを含まないようにしてください。

ODBC データ・ソースの作成については、『ASA データベース管理ガイド』> 「ODBC データ・ソースの作成」を参照してください。

パスワードを含む設定ファイルの暗号化

設定ファイルを作成するとき、パスワード情報をオプションとして組み込むことができます。パスワードを保護するために、ファイル非表示 (dbfhide) ユーティリティを使用し、簡単な暗号化で設定ファイルの内容を隠すことを検討してください。

ファイル非表示 (dbfhide) ユーティリティを使用して設定ファイルを非表示にする方法については、『ASA データベース管理ガイド』> 「dbfhide コマンド・ライン・ユーティリティを使用してファイル内容を隠す」を参照してください。

ユーザが実行できるタスクの制御

ユーザは、アクセスを付与されたオブジェクトにだけアクセスできます。

オブジェクトに対するパーミッションを別のユーザに付与するには、GRANT 文を使用します。あるオブジェクトに対するユーザ・パーミッションを他のユーザに渡して、付与することもできます。

GRANT 文は、より一般的なパーミッションをユーザに付与する場合にも使用します。ユーザに CONNECT パーミッションを付与すると、データベースに接続してパスワードを変更することができます。RESOURCE 権限が付与されたユーザは、テーブル、ビュー、プロシージャなどを作成できます。DBA 権限が付与されたユーザは、データベース内で何でも参照したり実行したりできます。DBA は、グループの作成と管理にも GRANT 文を使用します。

REVOKE 文は、GRANT 文と反対の機能を持っています。GRANT によって明示的に付与されたパーミッションは、REVOKE によって取り消されます。ユーザから CONNECT を取り消すと、そのユーザは、所有するすべてのオブジェクトとともにデータベースから削除されます。

ネガティブ・パーミッション

Adaptive Server Anywhere は「**ネガティブ・パーミッション**」をサポートしません。つまり、明示的に付与されていないパーミッションは取り消せません。

たとえば、ユーザ bob が sales グループのメンバであるとします。あるユーザがテーブル T に対する DELETE パーミッションを sales に付与すると、bob は T からローを削除できるようになります。bob に T からの削除を実行させないようにする場合は、REVOKE DELETE を単純に実行しただけでは bob から T へのパーミッションを取り消すことはできません。T に対する DELETE パーミッションは、直接 bob に付与されたものではないためです。この場合は、sales グループの bob のメンバシップを取り消さなければなりません。

詳細については、次の項目を参照してください。

- 『ASA SQL リファレンス・マニュアル』> 「GRANT 文」
- 『ASA SQL リファレンス・マニュアル』> 「REVOKE 文」

セキュリティを考慮したデータベース・オブジェクトの設計

ビューとストアド・プロシージャは、ユーザがアクセスできるデータと実行できるタスクをチューニングする代替方法を提供します。

これらの機能の詳細については、次の項目を参照してください。

- 『ASA SQL ユーザーズ・ガイド』> 「プロシージャとトリガの利点」
- 『ASA データベース管理ガイド』> 「高度なセキュリティを実現するためのビューとプロシージャの使い方」

データベース・アクティビティの監査

監査は、データベース上で行われたアクティビティをトラッキングする方法です。アクティビティの記録はトランザクション・ログに保持されます。監査を有効にすると、DBA はトランザクション・ログに保存するデータ量を増やして、次のデータも含みます。

- すべてのログイン試行 (成否とも)。ターミナル ID を含む。
- すべてのイベントの正確なタイムスタンプ (ミリ秒まで解析)。
- すべてのパーミッションの検査 (成否とも)。パーミッションが検査されたオブジェクトがあれば、それも含む。
- DBA 権限を必要とするすべてのアクション。

トランザクション・ログ

各データベースには、関連付けられたトランザクション・ログ・ファイルがあります。トランザクション・ログはデータベースのリカバリに使用します。これは、データベースに対して実行されたトランザクションの記録です。

トランザクション・ログの詳細については、『ASA データベース管理ガイド』> 「トランザクション・ログ」を参照してください。

トランザクション・ログには、実行されたすべてのデータ定義文と、それを実行したユーザ ID が格納されます。また、すべての更新、削除、挿入、これらの文を実行したユーザも格納されます。ただし、監査の目的によっては、これでは不十分です。デフォルトでは、トランザクション・ログにはイベントの時間は含まれず、イベントが発生した順序だけが含まれます。また、失敗したイベントや、SELECT 文も含まれません。

監査の有効化

データベース管理者が「**監査**」を有効にすると、セキュリティ関連の情報がトランザクション・ログに追加されます。

監査はデフォルトでは無効になっています。データベース上で監査を有効にするには、DBA でパブリック・オプション AUDITING の値を ON に設定します。設定すると、AUDITING オプションの値を OFF に設定して明示的に無効にするまで、監査は有効です。このプロシージャの設定には、DBA パーミッションが必要です。

❖ 監査を有効にするには、次の手順に従います。

1. データベースがバージョン 6.0.2 以上にアップグレードされていることを確認します。
2. データベースをアップグレードする必要がある場合は、新しいトランザクション・ログを作成します。
3. 次の文を実行します。

```
SET OPTION PUBLIC.AUDITING = 'ON'
```

詳細については、『ASA データベース管理ガイド』>「AUDITING オプション [データベース]」を参照してください。

監査情報の取り出し

ログ変換 (dbtran) ユーティリティを使用して、監査情報を取り出すことができます。このユーティリティは、Sybase Central またはコマンド・プロンプトからアクセスできます。これはトランザクション・ログに作用して、各コマンドを実行したユーザの情報と、すべてのトランザクションを保持する SQL スクリプトを生成します。*dbtran* は、-g オプションを使用することによって、監査情報を含むより多くのコメントを生成します。

監査レコードを完全で見やすいものにするため、-g オプションは次のオプションを自動的に設定します。

- **-d** 出力を日付順に表示します。
- **-t** トリガで生成したオペレーションを出力に含めます。
- **-a** ロールバック・トランザクションを出力に含めます。

ログ変換ユーティリティは、稼働中のデータベース・サーバに対して、またはデータベース・ログ・ファイルに対して実行できます。

❖ 稼働中のデータベース・サーバから監査情報を取り出すには、次の手順に従います。

1 ユーザ ID に DBA 権限があることを確認します。

2 データベース・サーバの稼働中に、システムのコマンド・プロンプトで次の文を実行します。

```
dbtran -g -c "uid=DBA;pwd=SQL;..." -n asademo.SQL
```

接続文字列の詳細については、『ASA データベース管理ガイド』> 「接続パラメータ」を参照してください。

❖ トランザクション・ログ・ファイルから監査情報を取り出すには、次の手順に従います。

1 データベース・サーバを閉じて、ログ・ファイルが使用可能であることを確認します。

2 システムのコマンド・プロンプトで次の文を実行して、ファイル *asademo.log* の情報をファイル *asademo.SQL* に格納します。

```
dbtran -g asademo.log
```

-g オプションによって、出力ファイルに監査情報が含まれます。

詳細については、『ASA データベース管理ガイド』> 「ログ変換ユーティリティ」を参照してください。

監査コメントの追加

sa_audit_string システム・ストアド・プロシージャを使用して、監査証跡にコメントを追加できます。これは引数を 1 つとります。引数は 200 バイト以内の文字列です。このプロシージャを呼び出すには、DBA パーミッションが必要です。

次に例を示します。

```
call sa_audit_string( 'Started audit testing here.' )
```

このコメントは監査文としてトランザクション・ログに格納されます。

監査の例

この例では、権限のない情報へのアクセス試行を、監査機能がどのようにして記録するかを示します。

1. データベース管理者として、監査を有効にします。

   これは、次のように Sybase Central から実行できます。

   - ASA 9.0 Sample データ・ソースに接続します。このとき **DBA** ユーザとして接続します。
   - **asademo** データベースのアイコンを選択し、[ファイル] メニューから [オプション] を選択します。
   - オプションのリストから [Auditing] を選択し、[値] ボックスに値 ON を入力します。[設定] をクリックしてオプションを設定し、[閉じる] をクリックして終了します。

   または、Interactive SQL を使用することもできます。ユーザ ID **DBA**、パスワード **SQL** を使って Interactive SQL からサンプル・データベースに接続し、次の文を実行します。

   ```
   SET OPTION PUBLIC.AUDITING = 'ON'
   ```

2. パスワードが **BadUser** のユーザ BadUser をサンプル・データベースに追加します。これは、Sybase Central から実行できます。または、Interactive SQL を使用して、次の文を入力することもできます。

   ```
   GRANT CONNECT TO BadUser
    IDENTIFIED BY 'BadUser'
   ```

3. Interactive SQL を使用して、BadUser としてサンプル・データベースに接続し、次に示すクエリで **employee** テーブルの機密情報にアクセスを試みます。

```
SELECT emp_lname, salary
 FROM DBA.employee
```

「employee から選択するためのパーミッションがありません。」というエラー・メッセージが表示されます。

4. コマンド・プロンプトからサンプル・データベースが保持されている Adaptive Server Anywhere インストール・ディレクトリに移動して、次のコマンドを実行します。

   ```
   dbtran -g -c "dsn=ASA 9.0 Sample" -n asademo.SQL
   ```

   このコマンドは、トランザクション・ログ情報と監査情報を持つ一連のコメントを含む *asademo.SQL* ファイルを生成します。権限のない BadUser による employee テーブルへのアクセス試行を示す行は、ファイルに次のように含まれています。

   ```
   --AUDIT-1001-0000287812 -- 2004/02/11 13:59:58.765
   Checking Select permission on employee - Failed
    --AUDIT-1001-0000287847 -- 2004/02/11 13:59:58.765
   Checking Select permission on employee(salary) -
   Failed
   ```

5. このマニュアル内で実行するその他の例が予期した結果になるように、サンプル・データベースを元の状態にリストアします。

   DBA ユーザとして接続し、次の操作を実行します。

   - ユーザ ID **BadUser** から接続の権限を取り消します。
   - PUBLIC.AUDITING オプションを OFF に設定します。

データベース・サーバ外のアクションの監査

データベース・ユーティリティの中には、データベース・ファイルに直接作用するものがあります。セキュリティを考慮した環境では、信用のおけるユーザ以外はデータベース・ファイルにアクセスできないようにしてください。

アクションの監査を実行するとき、Windows NT の場合にかぎっては、dbtran、dbwrite、dblog を使用して、データベース・ファイルと同じディレクトリに拡張子 *.alg* のテキスト・ファイルを生成します。たとえば、*asademo.db* の場合は、*asademo.alg* というファイルが生成さ

れます。ツール名、Windows ユーザ名、日付／時刻を含むレコードがこのファイルに追加されます。AUDITING オプションが ON に設定されている場合は、レコードが *.alg* ファイルに追加されるだけです。

安全な方法でのデータベース・サーバの実行

データベース・サーバ起動時、またはサーバのオペレーション中に設定できる、次のようなセキュリティ機能があります。

- **データベースの開始と停止** デフォルトでは、どのユーザでも、実行中のサーバで追加のデータベースを起動できます。-gd オプションは、この機能を実行できるユーザを、すでに接続しているデータベースで特定のレベルのパーミッションを付与されているユーザに制限できます。パーミッション値は **DBA**、**all**、または **none** です。

  詳細については、『ASA データベース管理ガイド』> 「-gd サーバ・オプション」を参照してください。

- **データベースの作成と削除** デフォルトでは、すべてのユーザが CREATE DATABASE 文を使用してデータベース・ファイルを作成できます。-gu オプションは、この機能を実行できるユーザを、すでに接続しているデータベースで特定のレベルのパーミッションを付与されているユーザに制限できます。パーミッション値は **DBA**、**all**、**none**、または **utility_db** です。

  詳細については、『ASA データベース管理ガイド』> 「-gu サーバ・オプション」を参照してください。

- **サーバの停止** dbstop ユーティリティは、データベース・サーバを停止します。このユーティリティは、バッチ・ファイルや、サーバを対話形式で停止 (サーバ・メッセージ・ウィンドウの [シャットダウン] をクリックして行う) できない場合に便利です。デフォルトでは、すべてのユーザが dbstop を実行してサーバを停止できます。-gk オプションは、この機能を実行できるユーザを、データベースで特定のレベルのパーミッションを付与されているユーザに制限できます。パーミッション値は **DBA**、**all**、または **none** です。

  詳細については、『ASA データベース管理ガイド』> 「-gk サーバ・オプション」を参照してください。

- **データのロードとアンロード** LOAD TABLE 文、UNLOAD TABLE 文、UNLOAD 文はすべて、データベース・サーバ・マシン上のファイル・システムにアクセスできます。パーソナル・

データベース・サーバが稼働中の場合、すでにファイル・システムにアクセスできるため、セキュリティ上の問題はありません。ネットワーク・データベース・サーバを稼働中の場合は、ファイル・システムへの不当なアクセスによってセキュリティ問題の起こる可能性があります。-gl オプションを使用して、データのロードとアンロードを行うのに必要なデータベース・パーミッションを制御できます。パーミッション値は **DBA**、**all**、または **none** です。

詳細については、『ASA データベース管理ガイド』> 「-gl サーバ・オプション」を参照してください。

- **トランスポート・レイヤ・セキュリティによるクライアント／サーバ通信の暗号化**　トランスポート・レイヤ・セキュリティを使用して、クライアント・アプリケーションとデータベース・サーバ間の通信を認証することで、ネットワーク・パケットのセキュリティを高めることができます。トランスポート・レイヤ・セキュリティでは、楕円曲線暗号方式または RSA 暗号方式を使用します。

  詳細については、「Adaptive Server Anywhere トランスポート・レイヤ・セキュリティ」33 ページを参照してください。

データベースの暗号化

データベース管理者として、データベースの暗号化を使用して第三者によるデータの解読を困難にできます。データベースを安全に管理するために、簡単な暗号化または高度な暗号化のいずれかを選択できます。

警告
暗号化されたデータベースを圧縮すると、データベースから暗号化が削除されます。

簡単な暗号化

簡単な暗号化は、難読化と同じです。これにより第三者は、ディスク・ユーティリティを使用してファイルを表示し、データベースのデータを解読することが困難になります。簡単な暗号化では、データベースの暗号化のためのキーは不要です。簡単な暗号化方式は、旧バージョンの SQL Anywhere Studio でサポートされています。

❖ 簡単な暗号化を使用するには、次の手順に従います。

- dbinit -e オプションを使用して、データベースを作成します。

  次の例では、簡単な暗号化を使用して、データベース *test.db* を作成します。

  ```
  dbinit -p 4096 -e test.db
  ```

  詳細については、『ASA データベース管理ガイド』> 「dbinit コマンド・ライン・ユーティリティを使用したデータベースの作成」を参照してください。

高度な暗号化

高度なデータベース暗号化方式では、キー (パスワード) がないとデータベースの操作やアクセスを行うことができません。アルゴリズムは、データベースやトランザクション・ログ・ファイルに含まれる情報にスクランブルをかけて解読できないようにしています。

警告
キーは保護してください。キーのコピーは、安全な場所に保管してください。キーを紛失すると、データベースにまったくアクセスできなくなり、リカバリも不可能になります。

AES は、Adaptive Server Anywhere の高度な暗号化を実装するために使用されているアルゴリズムです。これは、米国商務省標準技術局 (NIST：National Institute of Standards and Technology) によってブロック暗号のための新しい次世代標準暗号化方式 (AES：Advanced Encryption Standard) として選択されたブロック暗号化アルゴリズムです。

サポートされている 32 ビット Windows プラットフォームでは、AES_FIPS タイプを使用して、別個の FIPS 認定 AES アルゴリズムを高度な暗号化として指定することもできます。-fips オプションを指定してデータベース・サーバを起動する場合、AES または AES_FIPS の高度な暗号化方式で暗号化されたデータベースを実行できますが、簡単な暗号化方式で暗号化されたデータベースは実行できません。-fips を指定する場合、暗号化されていないデータベースをサーバ上で起動することもできます。

詳細については、『ASA データベース管理ガイド』> 「-fips サーバ・オプション」を参照してください。

AES_FIPS で暗号化されたデータベースを実行するマシンには、SQL Anywhere Studio セキュリティ・オプションをインストールしてください。

別途ライセンスを取得できるオプションが必要
AES_FIPS を使用する高度なデータベース暗号化では、別途ライセンスの SQL Anywhere Studio セキュリティ・オプションを入手する必要があります。このセキュリティ・オプションは、輸出規制対象品目です。

このコンポーネントのご注文については、『SQL Anywhere Studio の紹介』> 「別途ライセンスが入手可能なコンポーネント」を参照してください。

高度な暗号化を使用して新しいデータベースを作成するには、次の方法を使用できます。

- データベース初期化ユーティリティ (dbinit) と、高度な暗号化を有効にするための各種オプションの組み合わせ。

  dbinit ユーティリティと -ek オプションまたは -ep オプションを使用すると、高度な暗号化を使用するデータベースが作成され、プロンプト・ボックスまたはコマンド・ラインで暗号化キーを指定することができます。dbinit -ea オプションは、暗号化アルゴリズムを AES、または FIPS 認定アルゴリズムの場合は AES_FIPS に設定します。

  詳細については、『ASA データベース管理ガイド』>「初期化ユーティリティのオプション」および『ASA データベース管理ガイド』>「初期化ユーティリティ」を参照してください。

- CREATE DATABASE 文の ENCRYPTION 句。KEY オプションは暗号化キーを設定し、ALGORITHM オプションは暗号化アルゴリズムを AES、または FIPS 認定アルゴリズムの場合は AES_FIPS に設定します。

  また、Sybase Central の [データベース作成] ウィザードを使用して、高度に暗号化されたデータベースを作成することもできます。

  詳細については、『ASA SQL リファレンス・マニュアル』>「CREATE DATABASE 文」を参照してください。

- データベースのアンロード・ユーティリティ (dbunload) と、新規データベースを高度な暗号化で作成するためのオプション。-an オプションは、新規データベースを作成します。高度な暗号化と暗号化キーをプロンプト・ボックスまたはコマンド・ラインで指定するには、-ek オプションまたは -ep オプションを使用します。-ea オプションは、暗号化アルゴリズムを AES、または FIPS 認定アルゴリズムの場合は AES_FIPS に設定します。

  また、Sybase Central の [データベースのアンロード] ウィザードを使用して、高度に暗号化されたデータベースを作成することもできます。

詳細については、『ASA データベース管理ガイド』> 「アンロード・ユーティリティのオプション」および『ASA データベース管理ガイド』> 「アンロード・ユーティリティ」を参照してください。

❖ 高度に暗号化されたデータベースを作成するには、次の手順に従います (SQL の場合)。

1. Interactive SQL から既存のデータベースに接続します。
2. ENCRYPTION 句、KEY オプション、ALGORITHM オプションを含む CREATE DATABASE 文を実行します。

   たとえば、次の文は、FIPS 認定 AES 暗号化を使用して、*C:¥* ディレクトリにデータベース・ファイル *myencrypteddb.db* を作成します。

```
CREATE DATABASE 'c:¥¥myencrypteddb'
 TRANSACTION LOG ON
 ENCRYPTED ON
   KEY '0kZ2o52AK#'
   ALGORITHM 'AES_FIPS'
```

❖ 高度に暗号化されたデータベースを作成するには、次の手順に従います (コマンド・プロンプトの場合)。

1. コマンド・プロンプトで、dbinit ユーティリティを使用してデータベースを作成します。コマンド・プロンプトまたはダイアログ・ボックスで暗号化キーを指定するには、-ek または -ep をそれぞれ指定します。

   次のコマンドは、高度に暗号化されたデータベースを作成し、暗号化キーとアルゴリズムを指定します。

```
dbinit -ek "0kZ2o56AK#" -ea AES_FIPS
"myencrypteddb.db"
```

2. コマンド・プロンプトからデータベースを起動します。

```
dbeng9 myencrypteddb.db -ek "0kZ2o56AK#"
```

暗号化キーの詳細については、『ASA データベース管理ガイド』> 「DatabaseKey 接続パラメータ [DBKEY]」を参照してください。

ほとんどのパスワードと同様、最善の方法は、簡単には推測できないキー値を選択することです。キーには 8 ～ 30 文字の値を選択し、大文字と小文字、数字、文字、特殊文字を組み合わせて使用することをおすすめします。

警告
キーのコピーは、安全な場所に保管してください。キーは、データベースを起動したり変更したりするたびに必要になります。キーを紛失すると、データベースにまったくアクセスできなくなり、リカバリも不可能になります。

高度な暗号化の制御

Adaptive Server Anywhere では、データベース管理者が管理する高度な暗号化のテクノロジは 4 つあります。それは、高度な暗号化のステータス、暗号化キー、暗号化キーの保護、暗号化アルゴリズムです。

高度な暗号化のステータス

既存のデータベースでは高度な暗号化のオンとオフを簡単に切り替えることはできませんが、高度な暗号化の実装は、2 つのオプションから選択できます。高度な暗号化を指定して新規にデータベースを作成するか、既存のデータベースを再構築して同時に暗号化ステータスを変更するかのいずれかです。データベースの再構築では、既存のデータベースに含まれるすべてのデータとスキーマをアンロードし、新しいデータベースを作成して (ここで高度な暗号化のステータスを含めたさまざまな設定を変更できます)、データを新しいデータベースに再ロードします。高度に暗号化されたデータベースをアンロードするにはキーが必要です。

これらの機能の詳細については、次の項目を参照してください。

- 『ASA SQL ユーザーズ・ガイド』> 「データベースの再ロード」

- 『ASA SQL リファレンス・マニュアル』> 「CREATE DATABASE 文」

暗号化キー

ほとんどのパスワードと同様、最善の方法は、簡単には推測できないキー値を選択することです。キーの長さは任意ですが、短いと推測されやすいため、一般的には長い方が適しています。また、数字、文字、特殊文字を組み合わせると、キーは推測されにくくなります。データベースを起動するたびに、キーを指定してください。キーを忘れた場合はデータベースにまったくアクセスできなくなります。

暗号化キーの保護

暗号化キーの入力に、コマンド・プロンプト (デフォルト) またはプロンプト・ボックスのいずれかを選択できます。プロンプト・ボックスでのキー入力を選択すると、キーが表示されないため、さらにセキュリティが強化されます。クライアントでは、データベースを起動するたびにキーを指定してください。データベース管理者がデータベースを起動する場合は、クライアントでキーを使用する必要はありません。

詳細については、『ASA データベース管理ガイド』> 「-ep サーバ・オプション」を参照してください。

暗号化アルゴリズム

データベースを高度に暗号化する場合は、AES アルゴリズムを使用してデータベースが暗号化されます。

AES は、国際評価期間が最近終了し、新しい American Encryption Standard ブロック暗号化アルゴリズムとして選ばれました。これは、パフォーマンスやサイズの面で Adaptive Server Anywhere データベースの暗号化に役立つ多くのプロパティを備えています。サポートされている 32 ビット Windows プラットフォームでは、AES_FIPS アルゴリズムも使用できます。

データベース暗号化アルゴリズムの詳細については、以下の項目を参照してください。

- 『ASA データベース管理ガイド』> 「初期化ユーティリティのオプション」
- 『ASA SQL リファレンス・マニュアル』> 「CREATE DATABASE 文」

パフォーマンスの問題

Adaptive Server Anywhere のパフォーマンスは、データベースが暗号化されている場合にはある程度低下します。パフォーマンスの影響は、ディスクとのページの読み取りや書き込みの頻度によって異なります。また、サーバが使用するキャッシュ・サイズを適切に設定することによって影響を最小限にできます。

キャッシュの初期サイズを増やすには、サーバの起動時に -c オプションで指定します。キャッシュの動的なサイズ変更がサポートされているオペレーティング・システムでは、使用されるキャッシュ・サイズが、使用可能なメモリの容量によって制限される場合があります。そのため、キャッシュ・サイズを増加するには、使用可能なメモリを増加します。

詳細については、次の項目を参照してください。

- 『ASA SQL ユーザーズ・ガイド』> 「パフォーマンス向上へのキャッシュの使用」
- 『ASA データベース管理ガイド』> 「-c サーバ・オプション」

データベースの一部の暗号化

データベースの一部のみを暗号化する場合は、ENCRYPT 関数を使用して行います。ENCRYPT 関数は、同じ AES の高度な暗号化アルゴリズムを使用します。このアルゴリズムはデータベースの暗号化用に使用され、その関数に渡される値を暗号化します。

ENCRYPT 関数は、キーを使用して、渡される値を暗号化します。このキーは、大文字と小文字を区別しないデータベース内であっても、大文字と小文字が区別されます。ほとんどのパスワードと同様、最善の方法は、簡単には推測できないキー値を選択することです。キーには最低でも 16 文字の値を選択し、大文字と小文字、数字、文字、特殊文字を組み合わせて使用することをおすすめします。このキーは、データを復号化するたびに必要になります。

警告
キーは保護してください。キーのコピーは、安全な場所に保管してください。キーを紛失すると、暗号化データにまったくアクセスできなくなり、そこからのリカバリも不可能になります。

暗号化された値は、ENCRYPT 関数で指定したキーと同じキーを使用して、DECRYPT 関数で復号化できます。これらの関数はともに LONG BINARY 値を返します。異なるデータ型を使用する必要がある場合は、CAST 関数を使用して、その値を必要なデータ型に変換できます。次の例では、CAST 関数を使用して、復号化された値を必要なデータ型に変換する方法を示します。

CAST 関数の詳細については、『ASA SQL リファレンス・マニュアル』> 「CAST 関数 [データ型変換]」を参照してください。

データベース・ユーザが復号化された形式でデータをアクセスする必要があっても、暗号化キーにアクセスさせたくない場合は、DECRYPT 関数を使用したビューを作成できます。これにより、ユーザは暗号化キーを知らなくても、復号化されたデータにアクセスできるようになります。テーブルを使用したビューまたはストアド・プロシージャを作成する場合は、ALTER VIEW や ALTER PROCEDURES の SET HIDDEN パラメータを使用して、ユーザが暗号化キーをアクセスできないようにすることができます。

詳細については、『ASA SQL リファレンス・マニュアル』> 「ALTER PROCEDURE 文」と 『ASA SQL リファレンス・マニュアル』> 「ALTER VIEW 文」を参照してください。

カラムの暗号化の例

次の例では、user_info というテーブルのパスワードを格納するカラムを暗号化するトリガを使用します。user_info テーブルは、次のように定義されています。

```
CREATE TABLE user_info (
    emp_id INTEGER NOT NULL PRIMARY KEY,
    user_name CHAR(80),
    user_pwd CHAR(80) )
```

新しいユーザが追加されたとき、または既存のユーザのパスワードが更新されたときに、2 つのトリガが user_pwd カラムの値を暗号化するためにデータベースに追加されます。

- encrypt_new_user_pwd トリガは、新しいローが user_info テーブルに追加されるたびに実行されます。

```
CREATE TRIGGER encrypt_new_user_pwd
 BEFORE INSERT
 ON user_info
 REFERENCING NEW AS new_pwd
 FOR EACH ROW
 BEGIN
    SET new_pwd.user_pwd=ENCRYPT(new_pwd.user_pwd,
'8U3dkA');
 END
```

- encrypt_updated_pwd トリガは、user_info テーブルの user_pwd カラムが更新されるたびに実行されます。

```
CREATE TRIGGER encrypt_updated_pwd
 BEFORE UPDATE OF user_pwd
 ON user_info
 REFERENCING NEW AS new_pwd
 FOR EACH ROW
 BEGIN
    SET new_pwd.user_pwd=ENCRYPT(new_pwd.user_pwd,
'8U3dkA');
 END
```

データベースに新しいユーザを追加する場合

```
INSERT INTO user_info
 VALUES ( '1', 'd_williamson', 'abc123')
```

SELECT 文を発行して user_info テーブルの情報を表示する場合、user_pwd カラムの値はバイナリ・データ (パスワードの暗号化された形式) であり、INSERT 文で指定された値 **abc123** ではありません。

このユーザがパスワードを変更した場合

```
UPDATE user_info
 SET user_pwd='xyz'
 WHERE emp_id='1'
```

encrypt_updated_pwd トリガが実行され、新しいパスワードの暗号化された形式が user_pwd カラムに表示されます。

元のパスワードは、次の SQL 文を発行して検索できます。この文はデータを復号化するために DECRYPT 関数と暗号化キーを使用し、値を LONG BINARY から CHAR 値に変換するために CAST 関数を使用しています。

```
SELECT CAST (DECRYPT(user_pwd, '8U3dkA') AS CHAR(100))
FROM user_info
 WHERE emp_id = '1'
```

ENCRYPT および関数の詳細については、『ASA SQL リファレンス・マニュアル』>「アルファベット順の関数リスト」を参照してください。

Windows CE データベースの保護

ここでは、Windows CE データベースの安全管理に役立つ Adaptive Server Anywhere の機能について説明します。特に、監査とデータベース暗号化について説明します。他のセキュリティ機能の概要についても説明し、詳細情報の参照先も示しています。

データベース・ファイルの暗号化や簡単な通信暗号化など、Windows デスクトップ・プラットフォームを対象とする Adaptive Server Anywhere セキュリティ機能の多くは、Windows CE でもサポートされています。または、ログ変換ユーティリティのように、サポートが変更されているものもあります。

Windows CE 上で動作するデータベースは、Windows デスクトップ・プラットフォームで動作するデータベースと同じユーザ識別情報と認証機能を使用して、データベースにアクセスできるユーザと、そのユーザが実行できるアクションを制御します。

詳細については、「データベース・アクセスの制御」6 ページを参照してください。

Windows CE デバイス・セキュリティ

Windows CE デバイスに機密データを保存する場合は、Windows CE デバイス用に提供されているセキュリティ機能を使用できます。

使用できるセキュリティ機能の詳細については、Windows CE デバイスに付属しているユーザーズ・マニュアルを参照してください。

データベース・サーバ・オプション

サーバ・オプションを使用すると、サーバ上で特定の操作を実行できるユーザを制御できます。

このオプションは、Windows CE デバイス上でデータベースを起動するときに、[サーバ起動オプション] ダイアログの [オプション] フィールドで設定します。

詳細については、『ASA データベース管理ガイド』> 「コマンド・ラインからパーミッションを制御する」を参照してください。

Windows CE 上でのオプションの設定については、『SQL Anywhere Studio の紹介』> 「サーバとデータベースのオプション」を参照してください。

監査

この機能は、トランザクション・ログを使用して、データベース上でのアクションの詳細なレコードを管理します。

監査情報を含めて、トランザクション・ログに保存されている情報を変換するには、ログ変換ユーティリティ (dbtran) を使用します。Windows CE では dbtran ユーティリティがサポートされないため、Windows CE デバイスで保存されるログを変換することはできません。このユーティリティを使用するには、トランザクション・ログ・ファイルを PC にコピーします。

詳細については、『SQL Anywhere Studio セキュリティ・ガイド』>「データベース・アクティビティの監査」を参照してください。

Windows CE 上でのデータベースの暗号化

データベース暗号化機能を使用する際の、暗号化のレベルを選択します。データベースを安全に管理するために、簡単な暗号化または高度な暗号化のいずれかを選択できます。Adaptive Server Anywhere は、Windows CE 上で、簡単な暗号化と高度な暗号化の両方をサポートしています。

簡単な暗号化　このレベルの暗号化は、難読化と同じです。これにより第三者は、ディスク・ユーティリティを使用してファイルを表示し、データベースのデータを解読することが困難になります。簡単な暗号化では、データベースの暗号化のためのキーは不要です。

簡単な暗号化方式は、旧バージョンの SQL Anywhere Studio でサポートされています。

高度な暗号化　このレベルの暗号化は、データベースやトランザクション・ログ・ファイルに含まれる情報にスクランブルをかけることで、ディスク・ユーティリティを使用してファイルを表示するだけではデータを解読できないようにします。高度な暗号化にすると、キーなしではデータベースにまったくアクセスできなくなります。Windows CE 上で使用するデータベースを暗号化する場合は、AES アルゴリズムで暗号化してください。

詳細については、『SQL Anywhere Studio セキュリティ・ガイド』>「データベースの暗号化」を参照してください。

通信の暗号化と Windows CE

クライアント／サーバ通信を暗号化して、ネットワーク上の通信のセキュリティを強化できます。Adaptive Server Anywhere は、簡単な暗号化と高度な暗号化の、2 種類の通信暗号化を備えています。

簡単な通信暗号化は、簡単な暗号化を受けている通信パケットを受け取ります。このレベルの通信暗号化は、Windows CE と以前のバージョンの Adaptive Server Anywhere も含め、すべてのプラットフォームでサポートされます。

高度な通信暗号化は、Solaris、Linux、Mac OS X、NetWare、サポートされている 32 ビット Windows オペレーティング・システムの TCP/IP ポートでのみサポートされます。Windows CE では使用できません。

セキュリティのヒント

データベース管理者には、データのセキュリティ強化のために実行するアクションが多数用意されています。次に例を示します。

- **デフォルトのユーザ ID とパスワードの変更**　新しく作成されるデータベースのデフォルトのユーザ ID とパスワードは、**DBA** と **SQL** です。このパスワードを変更してから、データベースを展開してください。
- **長いパスワードが必要**　MIN_PASSWORD_LENGTH パブリック・オプションを設定して、短いパスワードを指定できないように (つまり簡単に推測できないように) することができます。

  詳細については、『ASA データベース管理ガイド』> 「MIN_PASSWORD_LENGTH オプション [データベース]」を参照してください。
- **DBA 権限の制限**　DBA 権限は非常に強力であるため、本当に必要なユーザにだけ付与するようにしてください。DBA 権限を持っているユーザは、データベース内で何でも参照したり実行したりできます。

  DBA 権限を持つユーザには、ユーザ ID を 2 つ与えてください。1 つを DBA 権限付き、もう 1 つを DBA 権限なしにすれば、必要なときにだけ DBA として接続できます。
- **外部システム関数の削除**　外部関数の xp_cmdshell、xp_startmail、xp_startsmtp、xp_sendmail、xp_stopmail、xp_stopsmtp は、セキュリティを考慮した環境で使用すると問題が発生する可能性があります。

  xp_cmdshell プロシージャを使用すると、ユーザはオペレーティング・システム・コマンドやプログラムを実行できます。

  e-mail コマンドを使用すると、ユーザは、ユーザが作成した e-mail をサーバで送信することができます。多数のユーザが、e-mail またはコマンド・シェル・プロシージャを使用して、付与されていない権限でオペレーティング・システムのタスクを実行できるようになります。セキュリティを考慮した環境では、このような関数は削除してください。

プロシージャの削除については、『ASA SQL リファレンス・マニュアル』> 「DROP 文」を参照してください。

- **データベース・ファイルの保護** データベース・ファイル、ログ・ファイル、DB 領域ファイル、ライト・ファイルは、権限のないアクセスを受け付けないようにしてください。このようなファイルは共有ディレクトリまたはボリュームには保管しないでください。

- **データベース・ソフトウェアの保護** Adaptive Server Anywhere ソフトウェアも同様に保護してください。ユーザには、アプリケーション、DLL、その他の必要なリソースへのアクセス権だけを付与してください。

- **サービスまたはデーモンとしてのデータベース・サーバの実行** 権限のないユーザがサーバを停止したり、データベースやログ・ファイルへのアクセスを取得したりしないように、データベース・サーバを Windows サービスとして実行してください。UNIX では、同様の目的でサーバをデーモンとして実行します。

  詳細については、『ASA データベース管理ガイド』> 「現在のセッション外でのサーバの起動」を参照してください。

- **ASTMP をユニーク・ディレクトリに設定** UNIX プラットフォームでエンジンを保護するには、ASTMP をユニークなディレクトリに設定し、ディレクトリの読み取り、書き込み、実行を他のすべてのユーザから保護します。これによって、すべての接続が強制的に TCP/IP を使用することになり、共有メモリ接続よりも安全になります。

- **データベースを高度に暗号化** データベースを高度に暗号化すると、キーがなければまったくアクセスできなくなります。他の方法を使っても、データベースを開いたり、データベースやトランザクション・ログ・ファイルを表示したりできません。

  詳細については、『ASA データベース管理ガイド』> 「-ep サーバ・オプション」と 『ASA データベース管理ガイド』> 「-ek データベース・オプション」を参照してください。

第 2 章

Adaptive Server Anywhere トランスポート・レイヤ・セキュリティ

この章の内容

この章では、Adaptive Server Anywhere データベース・サーバとクライアント・アプリケーション間の通信を、トランスポート・レイヤ・セキュリティを使用して保護する方法について説明します。

Mobile Link トランスポート・レイヤ・セキュリティの詳細については、『Mobile Link 管理ガイド』> 「Mobile Link トランスポート・レイヤ・セキュリティ」を参照してください。

Adaptive Server Anywhere Web サーバを設定してトランスポート・レイヤ・セキュリティを使用する方法については、『SQL Anywhere Studio セキュリティ・ガイド』> 「Web サービスでのトランスポート・レイヤ・セキュリティの使用」を参照してください。

個別にライセンスを取得可能なオプションが必要

トランスポート・レイヤ・セキュリティを使用するには、個別にライセンスを取得可能な SQL Anywhere Studio セキュリティ・オプションを入手する必要があります。このセキュリティ・オプションは、輸出規制対象品目です。

このコンポーネントの注文方法については、『SQL Anywhere Studio の紹介』> 「別途ライセンスが入手可能なコンポーネント」を参照してください。

概要

個別にライセンスを取得可能なオプションが必要
トランスポート・レイヤ・セキュリティを使用するには、個別にライセンスを取得可能な SQL Anywhere Studio セキュリティ・オプションを入手する必要があります。このセキュリティ・オプションは、輸出規制対象品目です。

このコンポーネントの注文方法については、『SQL Anywhere Studio の紹介』> 「別途ライセンスが入手可能なコンポーネント」を参照してください。

トランスポート・レイヤ・セキュリティは IETF 標準プロトコルであり、デジタル証明書とパブリック・キー暗号方式を使用して、クライアント／サーバ・アプリケーションを保護します。

クライアントは信用されたパブリック証明書を使用してデータを暗号化し、最初のクライアント／サーバ・ハンドシェイクでサーバを認証します。クライアントから送られてくるデータは、データベース・サーバ証明書に保存されている、対応するプライベート・キーによってのみ複号化できます。

サーバ認証の場合は、データベース・サーバが、そのパブリック証明書をクライアントに送信します。クライアントは、証明書のフィールドと証明書に埋め込まれているデジタル署名を使用して、サーバのアイデンティティを確認します。

効率性

トランスポート・レイヤ・セキュリティ標準は、パブリック・キー暗号方式に伴う非効率性を上回ります。安全な接続が確立されると、クライアントとサーバは共通キーを交換します。以降の通信では、非常に効率的な対称暗号が使用されます。

サポートされるプラットフォーム

トランスポート・レイヤ・セキュリティを使用するには、Solaris、Linux、NetWare、Mac OS X、またはサポートされている 32 ビットの Windows プラットフォーム (Windows CE 以外) で、サーバとクライアントの両方を実行してください。また、接続には TCP/IP ポートを使用します。

FIPS 承認のセキュリティ・オプションは Windows でのみ使用できます。

FIPS 140-2 承認

連邦情報処理規格 (FIPS) 140-2 では、セキュリティ・アルゴリズムの要件を指定しています。ただし、SSL などのセキュリティ・プロトコルやトランスポート・レイヤ・セキュリティの要件は指定していません。FIPS 140-2 は、米国商務省標準技術局 (NIST：National Institute of Standards and Technology) およびカナダ通信安全保障局 (CSE：Canadian Communication Security Establishment) を通じて、米国政府とカナダ政府から付与されます。Certicom は、Windows で実装されるセキュリティ・アルゴリズムについて、FIPS 証明を取得しています。

SQL Anywhere Studio が提供するトランスポート・レイヤ・セキュリティでは、Certicom ソフトウェア内の基本となる FIPS 承認アルゴリズムを使用できます。

トランスポート・レイヤ・セキュリティを使用するには、セキュリティ・オプションを別途購入する必要があります。

トランスポート・レイヤ・セキュリティの注文方法については、『SQL Anywhere Studio の紹介』> 「別途ライセンスが入手可能なコンポーネント」を参照してください。

FIPS 承認のセキュリティ・アルゴリズムを使用すると、データベース・ファイルを暗号化したり、データベース・クライアント／サーバ通信、Web サービス、Mobile Link クライアント／サーバ通信における通信を暗号化できます。

詳細については、次の項目を参照してください。

- 『SQL Anywhere Studio セキュリティ・ガイド』> 「データベースの暗号化」
- 『SQL Anywhere Studio セキュリティ・ガイド』> 「Web サービスでのトランスポート・レイヤ・セキュリティの使用」
- 「トランスポート・レイヤ・セキュリティを使用する データベース・サーバの起動」47 ページ
- 「トランスポート・レイヤ・セキュリティを使用する クライアント・アプリケーションの設定」49 ページ

トランスポート・レイヤ・セキュリティの設定

Adaptive Server Anywhere トランスポート・レイヤ・セキュリティを設定するには、次の手順に従います。

- **デジタル証明書を作成する** パブリック証明書とサーバ証明書を作成します。パブリック証明書がクライアント・アプリケーションに配布されるのに対して、サーバ証明書はデータベース・サーバに安全に保存されます。

  「デジタル証明書の作成」37 ページを参照してください。

- **トランスポート・レイヤ・セキュリティを指定して Adaptive Server Anywhere データベース・サーバを起動する** -ec データベース・サーバ・オプションを使用して、セキュリティのタイプ、サーバ証明書、プライベート・キーを保護するパスワードを指定します。

  『SQL Anywhere Studio セキュリティ・ガイド』> 「トランスポート・レイヤ・セキュリティを使用するデータベース・サーバの起動」を参照してください。

- **トランスポート・レイヤ・セキュリティを使用するようにクライアント・アプリケーションを設定する** Encryption 接続パラメータ [ENC] を使用して、信頼できるパブリック証明書のパスとファイル名を指定します。

  『SQL Anywhere Studio セキュリティ・ガイド』> 「トランスポート・レイヤ・セキュリティを使用するクライアント・アプリケーションの設定」を参照してください。

デジタル証明書の作成

トランスポート・レイヤ・セキュリティを設定するには、デジタル証明書を生成します。

自己署名証明書の作成、エンタープライズ・ルート証明書と証明書チェーンの使用、認証局 (CA) による証明書への署名を行うことができます。

- **自己署名証明書** 自己署名証明書は、単一のデータベース・サーバで構成される単純な設定において使用されます。この場合、信頼できるパブリック証明書を作成するために使用されたプライベート・キーは、民間の認証局や専用の設備ではなく、データベース・サーバに保存されます。

  「自己署名ルート証明書」38 ページを参照してください。

- **エンタープライズ・ルート証明書** エンタープライズ・ルート証明書を使用すると、複数のサーバが配備されている環境での、データの整合性と拡張性が向上します。

  - 信頼できるパブリック証明書を作成するために使用されるプライベート・キーは、安全な中央のロケーションに保存できます。
  - クライアントを再設定することなく、データベース・サーバを追加できます。

  「証明書チェーン」39 ページを参照してください。

- **民間の認証局** エンタープライズ・ルート証明書の代わりに、サードパーティの認証局を使用できます。民間の認証局は、プライベート・キーを保存するための専用の設備を備えており、高品質なサーバ証明書を作成します。

  詳細については、「証明書チェーン」39 ページおよび「グローバル署名証明書」42 ページを参照してください。

証明書ユーティリティ

SQL Anywhere Studio の証明書生成ユーティリティ gencert を使用すると、証明書を作成できます。このユーティリティでは、証明書識別情報とファイル情報の入力が要求され、RSA または楕円曲線暗号化方式が使用されます。証明書読み込みユーティリティ readcert を使用すると、証明書の値を表示し、証明書チェーンを検証できます。

自己署名ルート証明書

自己署名ルート証明書は、単一のデータベース・サーバで構成される単純な設定において使用されます。この場合、信頼できるパブリック証明書を作成するために使用されたプライベート・キーは、民間の認証局や専用の設備ではなく、データベース・サーバに保存されます。

ヒント
複数のデータベース・サーバを操作している場合、または証明書の高度な整合性が要求される場合は、エンタープライズ・レベルの証明書チェーンを使用します。

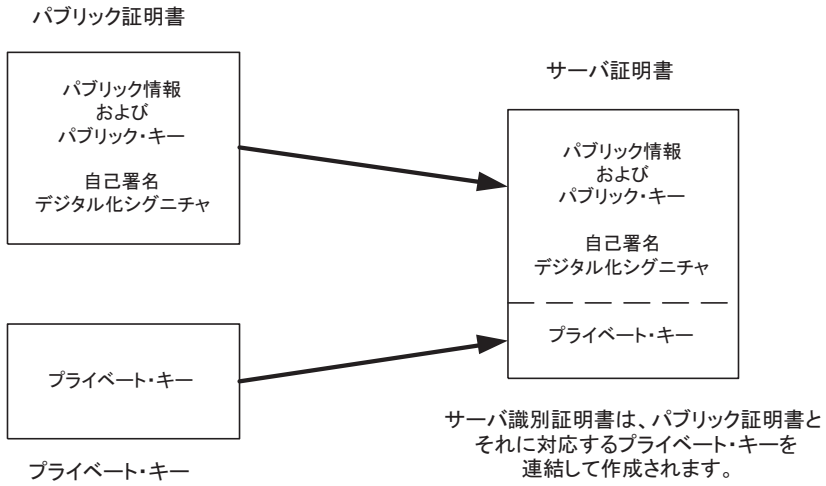
証明書チェーンの設定については、「証明書チェーン」39 ページを参照してください。

自己署名証明書を設定するには、gencert ユーティリティを使用して次の証明書を生成します。

- **パブリック証明書** 自己署名パブリック証明書は、クライアント・アプリケーションに配布されます。これは、識別情報、データベース・サーバのパブリック・キー、サーバ認証用の自己署名デジタル署名が含まれる電子文書です。

  詳細については、『SQL Anywhere Studio セキュリティ・ガイド』> 「トランスポート・レイヤ・セキュリティを使用するデータベース・サーバの起動」を参照してください。

- **サーバ証明書** サーバ証明書は、データベース・サーバに安全に保存されます。これは、自己署名パブリック証明書 (クライアントに配布) と、対応するプライベート・キーを組み合わせたものです。プライベート・キーによって、データベース・サーバは、クライアント・アプリケーションから送信されてくるメッセージを復号化できます。



自己署名ルート証明書を生成する方法については、『Mobile Link 管理ガイド』>「証明書生成ユーティリティ」を参照してください。

証明書チェーン

自己署名証明書ではなく、証明書チェーンを使用すると、マルチサーバ環境のセキュリティと拡張性を向上できます。証明書チェーンでは、認証局またはエンタープライズ・ルート証明書による、データベース・サーバ証明書への署名が必要です。

自己署名証明書の詳細については、「自己署名ルート証明書」38 ページを参照してください。

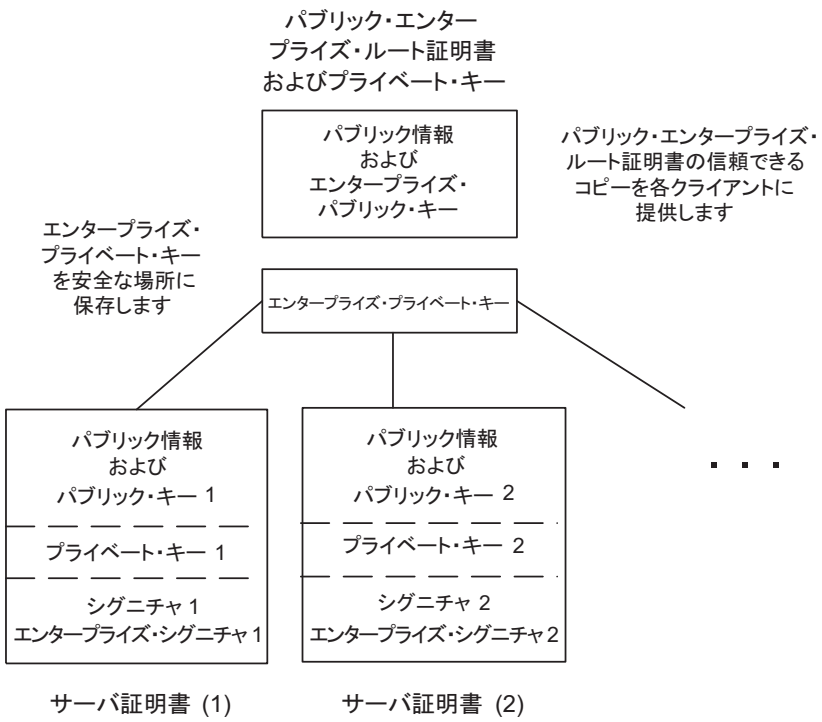
証明書チェーンを使用する利点

証明書チェーンには、次の利点があります。

- **拡張性** クライアント・アプリケーションが、エンタープライズ・ルート証明書または認証局によって署名されたすべての証明書を信用するように設定できます。新しいデータベース・サーバを追加しても、クライアントは新しいパブリック証明書のコピーを必要としません。

- **セキュリティ** エンタープライズ・ルート証明書のプライベート・キーは、データベース・サーバにはありません。ルート証明書のプライベート・キーを高セキュリティのロケーションに保存したり、専用の設備を備えている認証局を使用することで、サーバ認証の整合性が保護されます。

次の図は、エンタープライブ・ルート証明書の基本アーキテクチャを示しています。



マルチサーバ環境で使用する証明書を作成するには、次の手順に従います。

- パブリック・エンタープライズ・ルート証明書およびエンタープライズ・プライベート・キーを生成します。

パブリック・エンタープライズ・ルート証明書は、クライアント・アプリケーションに配布します。エンタープライズ・プライベート・キーは安全なロケーションに保存します。専用の設備の方が安全です。

- データベース・サーバごとにサーバ証明書を生成します。

  パブリック・エンタープライズ・ルート証明書とエンタープライズ・プライベート・キーを使用して、各サーバ証明書に署名します。

サードパーティの認証局を使用して、サーバ証明書に署名することもできます。民間の認証局は、プライベート・キーを保存するための専用の設備を備えており、高品質なサーバ証明書を作成します。

詳細については、「グローバル署名証明書」42 ページを参照してください。

エンタープライズ・ルート証明書

エンタープライズ・ルート証明書を使用すると、複数のサーバが配備されている環境での、データの整合性と拡張性が向上します。

- 信頼できるパブリック証明書を作成するために使用されるプライベート・キーは、安全な中央のロケーションに保存できます。
- クライアントを再設定することなく、データベース・サーバを追加できます。

エンタープライズ・ルート証明書を設定するには、エンタープライズ・ルート証明書と、サーバ証明書に署名するときに使用するエンタープライズ・プライベート・キーを作成します。

サーバ証明書の作成については、「署名付きのサーバ証明書」42 ページを参照してください。

エンタープライズ・ルート証明書を生成する方法については、『Mobile Link 管理ガイド』>「証明書生成ユーティリティ」を参照してください。

署名付きのサーバ証明書

データベース・サーバごとにサーバ証明書を生成します。これらの証明書はエンタープライズ・ルート証明書の署名を受けるため、gencert -s オプションを使用します。

署名付きサーバ証明書の生成については、『Mobile Link 管理ガイド』>「証明書生成ユーティリティ」を参照してください。

データベース・サーバごとに署名付きサーバ証明書を生成する方法については、『Mobile Link 管理ガイド』>「証明書生成ユーティリティ」を参照してください。

グローバル署名証明書

民間の認証局とは、高品質の証明書の作成と、これらの証明書を使用した証明書要求への署名を事業としている組織です。

グローバル署名証明書には、次の利点があります。

- 会社内での通信の場合、共通して信用するものとして、外部の認可された認証局を使用すると、システムのセキュリティの信頼性が高まります。認証局は、署名を行ったすべての証明書の識別情報が正確であることを保証する必要があります。
- 認証局は、証明書を生成するための管理された環境と高度な方法を提供します。
- ルート証明書のプライベート・キーは、秘密にしておきます。企業内ではこの重要情報を格納するのに適した場所がない可能性がありますが、認証局では専用の設備を設計して管理できます。

グローバル署名証明書の設定

グローバル署名証明書を設定するには、次の手順に従います。

- Certicom の reqtool ツールを使用して、証明書要求を作成します。

  「reqtool を使用してグローバル証明書を取得する」43 ページを参照してください。

- 認証局を使用して、各データベース・サーバの証明書要求に署名します。

「グローバル証明書をサーバ証明書として使用する」44 ページを参照してください。

エンタープライズ・ルート証明書へのグローバル署名
エンタープライズ・ルート証明書にグローバル署名することができます。これは、認証局が、他の証明書に署名できる証明書を生成する場合のみ適用されます。

reqtool を使用してグローバル証明書を取得する

Adaptive Server Anywhere トランスポート・レイヤ・セキュリティは、Certicom SSL/TLS Plus ライブラリを基にしています。このライブラリは、楕円曲線証明書または RSA 証明書が必要です。グローバル証明書は正しいフォーマットで証明書を提供する認証局から入手できます。

証明書を取得するには、いくつかの方法があります。たとえば、reqtool ユーティリティを使用して取得できます。このユーティリティは、セキュリティ・コンポーネントのインストール時にインストールされます。このツールはサーバのプライベート・キーとグローバル証明書要求を作成します。

例

次の例では、楕円曲線証明書要求を作成します。

```
> reqtool
 -- Certicom Corp. Certificate Request Tool 3.0d1 --
 Choose certificate request type:
   E - Personal email certificate request.
   S - Server certificate request.
   Q - Quit.
 Please enter your request [Q] : S
 Choose key type:
   R - RSA key pair.
   D - DSA key pair.
   E - ECC key pair.
   Q - Quit.
 Please enter your request [Q] : E
 Using curve ec163a02. Generating key pair (please
wait)...
 Country: CA
 State: Ontario
```

```
Locality: Waterloo
Organization: Sybase, Inc.
Organizational Unit: IAS
Common Name: IAS_Waterloo
Enter password to protect private key : mypwd123
Enter file path to save request : global.req
Enter file path to save private key :
serv1_private_key.pri
```

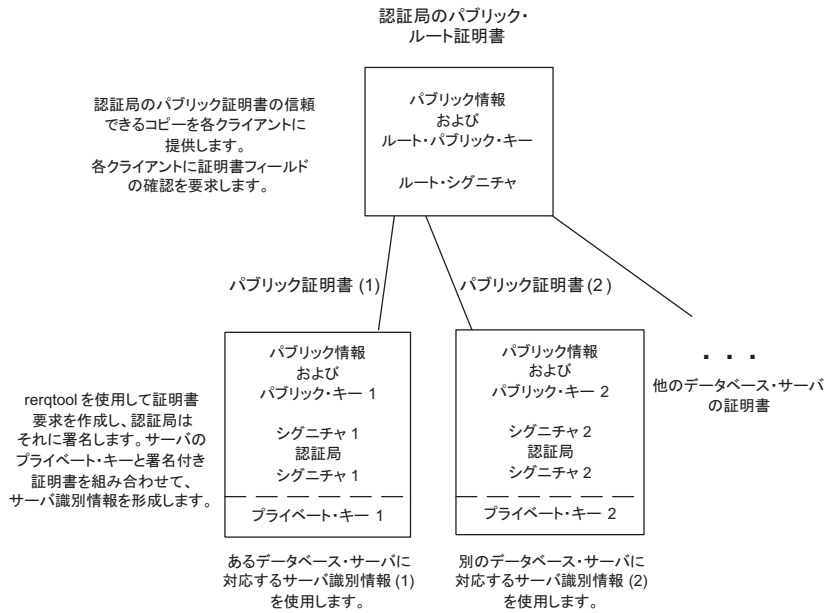
ファイル *global.req* には、パブリック証明書と要求情報があります。このファイルの内容を、証明書を発行している Web サイトのフォームにコピーします。認証局は、要求に署名し、パブリック証明書 *global.crt* を作成します。

ファイル *serv1_private_key.pri* には、対応するプライベート・キーが含まれています。このファイルは入力したパスワードで保護されますが、パスワードによる保護は弱いものなので、このファイルを安全なロケーションに格納してください。

reqtool の使用の詳細については、SQL Anywhere 9 インストール環境の *win32* サブディレクトリにあるマニュアル *reqtool.pdf* を参照してください。

グローバル証明書をサーバ証明書として使用する

グローバル署名証明書を直接、データベース・サーバ証明書として使用できます。次の図は、マルチサーバ環境の設定を示しています。



サーバ識別情報を作成するには、認証局の署名を受けたパブリック証明書と、reqtool ユーティリティを使用して作成したプライベート・キーを連結します。

reqtool ユーティリティの詳細については、「reqtool を使用してグローバル証明書を取得する」43 ページを参照してください。

次の例では、グローバル署名パブリック証明書 *global.crt* とプライベート・キー *serv1_private_key.pri* を連結して、サーバ証明書 *server1_certificate.crt* を作成します。

```
copy global.crt+serv1_private_key.pri
server1_certificate.crt
```

dbsrv9 コマンド・ラインで、サーバ証明書 *server1_certificate.crt* と、プライベート・キー *serv1_private_key.pri* のパスワードを参照します。

詳細については、『SQL Anywhere Studio セキュリティ・ガイド』>「トランスポート・レイヤ・セキュリティを使用するデータベース・サーバの起動」を参照してください。

認証局のパブリック証明書を信用するようにクライアントを設定する

データベース・サーバにアクセスするクライアントが、チェーン内のルート証明書を信用することを確認する必要があります。グローバル署名証明書の場合、ルート証明書は認証局のパブリック証明書です。

証明書フィールドの確認
グローバル署名証明書を使用する場合、各クライアント・アプリケーションはフィールド値を確認して、同じ認証局が他のクライアント用に署名した証明書を信用することを避けなければなりません。

サーバ証明書を信用するようにクライアント・アプリケーションを設定する方法については、『SQL Anywhere Studio セキュリティ・ガイド』>「トランスポート・レイヤ・セキュリティを使用するクライアント・アプリケーションの設定」を参照してください。

グローバル署名証明書を使用して信用を確立する方法については、「グローバル署名証明書」42 ページを参照してください。

トランスポート・レイヤ・セキュリティを使用するデータベース・サーバの起動

トランスポート・レイヤ・セキュリティを使用してデータベース・サーバを起動するには、サーバ証明書と、サーバのプライベート・キーを保護するパスワードを指定します。Adaptive Server Anywhere トランスポート・レイヤ・セキュリティは、TCP/IP と、Solaris、Linux、NetWare、Mac OS X、または Windows CE 以外のサポートされている任意の Windows プラットフォームでのみ使用できます。

トランスポート・レイヤ・セキュリティを設定する手順の概要については、「トランスポート・レイヤ・セキュリティの設定」36 ページを参照してください。

-ec サーバ・オプションを使用して、証明書と certificate_password パラメータを指定します。

次に示すのは、dbsrv9 コマンド・ラインの一部です。

-ec *cipher* **(certificate=***server-certificate***;certificate_password=***password***)**
-x tcpip

- ***cipher*** RSA 暗号化の場合は **rsa_tls** を指定し、楕円曲線暗号化の場合は **ecc_tls** をそれぞれ指定します。FIPS 承認の RSA 暗号化の場合は、**rsa_tls_fips** を指定します。rsa_tls_fips は別の承認ライブラリを使用しますが、Adaptive Server Anywhere 9.0.2 以降の、rsa_tls を指定しているクライアントと互換性があります。rsa_tls_fips 暗号化は、サポートされている 32 ビット Windows プラットフォームでのみ使用できます。

  cipher は、証明書を作成するときに使用される暗号化 (ECC または RSA) と一致する必要があります。

  FIPS 承認のアルゴリズムの実行については、『ASA データベース管理ガイド』>「-fips サーバ・オプション」を参照してください。

- ***server-certificate*** サーバ証明書のパスとファイル名を指定します。FIPS 承認の RSA 暗号化を使用している場合は、RSA 暗号化を使用して証明書を生成する必要があります。

サーバ証明書の作成については、「デジタル証明書の作成」37 ページを参照してください。サーバ証明書は、自己署名証明書、または認証局やエンタープライズ・ルート証明書の署名を受けた証明書のいずれかです。

- ***password*** サーバ証明書のプライベート・キーのパスワードを指定します。このパスワードは、サーバ証明書を作成するときに指定します。

単純暗号化を使用してデータベース・サーバを起動することもできますが、データの整合性は保証されず、サーバ認証を行うこともできません。単純暗号化を使用すると、パケット・スニッファを使用して、クライアントとサーバの間で送信されるネットワーク・パケットを読み取るのが困難になります。単純暗号化は、旧バージョンの SQL Anywhere Studio でサポートされています。

-ec サーバ・オプションの詳細については、『ASA データベース管理ガイド』> 「-ec サーバ・オプション」を参照してください。

TCP/IP プロトコルは、-x サーバ・オプションを使用して指定します。

詳細については、『ASA データベース管理ガイド』> 「-x サーバ・オプション」を参照してください。

例

次の例では、-ec サーバ・オプションを使用して、ecc_tls セキュリティ、サーバ証明書、サーバのプライベート・キーを保護するパスワードを指定します。

```
dbsrv9 -ec ecc_tls(certificate=c:¥test¥serv1_ecc.crt;
certificate_password=mypwd) -x tcpip asademo.db
```

設定ファイルとファイル非表示ユーティリティ dbfhide を使用して、passwords を含むコマンド・ライン・オプションを非表示にすることができます。詳細については、『ASA データベース管理ガイド』> 「@data サーバ・オプション」を参照してください。

次の例では、-ec サーバ・オプションを使用して、sa_tls セキュリティ、サーバ証明書、サーバのプライベート・キーを保護するパスワードを指定します。

```
dbsrv9 -ec rsa_tls(certificate=c:¥test¥serv1_rsa.crt;
certificate_password=test) -x tcpip asademo.db
```

トランスポート・レイヤ・セキュリティを使用するクライアント・アプリケーションの設定

トランスポート・レイヤ・セキュリティを使用するように、クライアント・アプリケーションを設定することができます。暗号化接続パラメータ・セットを使用して、信用されたパブリック証明書、暗号化のタイプ、ネットワーク・プロトコルを指定します。

トランスポート・レイヤ・セキュリティを設定する手順の概要については、「トランスポート・レイヤ・セキュリティの設定」36 ページを参照してください。

サーバ認証

リモート・クライアントは、サーバ認証を使用することで、データベース・サーバのアイデンティティを確認できます。デジタル署名と証明書フィールドの確認が一緒に機能して、サーバ認証が実現します。

デジタル署名

データベース・サーバ証明書には、データの整合性を維持し、不正侵入を防ぐための、複数のデジタル署名が含まれています。デジタル署名の作成は、次の手順で行われます。

- 証明書で実行されるアルゴリズムが、ユニークな値またはハッシュを生成します。
- 証明書への署名または認証局のプライベート・キーを使用して、ハッシュが暗号化されます。
- デジタル署名と呼ばれる暗号化ハッシュが、証明書に埋め込まれます。

デジタル署名は、自己署名、あるいはエンタープライズ・ルート証明書または認証局の署名を受けています。

クライアント・アプリケーションがデータベース・サーバにアクセスする場合、各クライアントがトランスポート・レイヤ・セキュリティを使用するように設定されていると、サーバはそのパブリック証明書

のコピーをクライアントに送信します。クライアントは、証明書に含まれているサーバのパブリック・キーを使用して証明書のデジタル署名を復号化し、証明書の新しいハッシュを算出して、2 つの値を比較します。値が一致する場合は、サーバの証明書の整合性が確認されます。

FIPS 承認の RSA 暗号化を使用している場合は、RSA を使用して証明書を生成する必要があります。

自己署名証明書の詳細については、「自己署名ルート証明書」38 ページを参照してください。

エンタープライズ・ルート証明書と認証局の詳細については、「証明書チェーン」39 ページを参照してください。

証明書フィールドの確認

グローバル署名証明書を使用する場合、各クライアントは証明書のフィールド値を確認して、同じ認証局が他のクライアント用に署名した証明書を信用することを避けなければなりません。この問題を解決するには、証明書の識別情報部分のフィールド値をテストするようにクライアントに要求します。認証局は、署名を行ったすべての証明書の識別情報が正確であることを保証する必要があります。

グローバル署名証明書の詳細については、「グローバル署名証明書」42 ページを参照してください。

gencert ユーティリティを使用して証明書を作成する場合は、組織、組織単位、通称のフィールドに値を入力します。対応するクライアント接続パラメータを使用して、これらのフィールドを確認します。

- **組織** 組織フィールドは、certificate_company 暗号化接続パラメータに対応します。
- **組織単位** 組織単位フィールドは、certificate_unit 暗号化接続パラメータに対応します。
- **通称** 通称フィールドは、certificate_name 暗号化接続パラメータに対応します。

クライアント側の暗号化接続パラメータの詳細については、『ASA データベース管理ガイド』> 「Encryption 接続パラメータ [ENC]」を参照してください。

クライアント・セキュリティ・オプション

クライアントは、トランスポート・レイヤ・セキュリティに対して暗号化接続パラメータのセットを使用します。

trusted_certificates オプション

このオプションだけが必須です。クライアントは、trusted_certificates 暗号化オプションを使用して、信用されたデータベース・サーバ証明書を指定します。信用された証明書は、サーバの自己署名パブリック証明書、パブリック・エンタープライズ・ルート証明書、民間の証明局に属するパブリック証明書のいずれかです。

詳細については、「デジタル証明書の作成」37 ページを参照してください。

証明書フィールドの確認

証明書フィールドを確認するには、certificate_company、certificate_unit、certificate_name の各暗号化プロトコル・オプションを使用します。これは、サーバ認証の重要な手順です。サードパーティの認証局を使用して証明書にグローバル署名している場合は、証明書フィールドを確認してください。

証明書フィールドの確認の詳細については、『SQL Anywhere Studio セキュリティ・ガイド』> 「証明書フィールドの確認」を参照してください。

トランスポート・レイヤ・セキュリティを使用するクライアント接続の確立

クライアント・アプリケーションがトランスポート・レイヤ・セキュリティを使用するように設定するには、接続文字列の中で Encryption (ENC) 接続パラメータを使用します。接続文字列の形式は次のとおりです。

Encryption=*cipher***(trusted_certificates=***public-certificate***)**

- ***cipher*** RSA 暗号化の場合は **rsa_tls** を指定し、楕円曲線暗号化の場合は **ecc_tls** をそれぞれ指定します。FIPS 承認の RSA 暗号化の場合は、**rsa_tls_fips** を指定します。rsa_tls_fips は別の承認ライブラリを使用しますが、Adaptive Server Anywhere 9.0.2 以降の、rsa_tls を指定しているサーバと互換性があります。rsa_tls_fips 暗号化は、サポートされている 32 ビット Windows プラットフォームでのみ使用できます。

  cipher に指定する暗号化が、証明書を作成するときに使用した暗号化 （RSA または ECC) と一致しない場合、接続は失敗します。

- ***public-certificate*** 信用されたパブリック証明書のパスとファイル名を指定します。FIPS 承認の RSA 暗号化を使用している場合は、RSA を使用して証明書を生成する必要があります。

trusted_certificates および他のクライアント・セキュリティ・パラメータの詳細については、『SQL Anywhere Studio セキュリティ・ガイド』> 「クライアント・セキュリティ・オプション」を参照してください。

パブリック証明書の作成または取得の詳細については、「デジタル証明書の作成」37 ページを参照してください。

暗号化接続パラメータの詳細については、『ASA データベース管理ガイド』> 「Encryption 接続パラメータ [ENC]」を参照してください。

例

次の例では、trusted_certificates 暗号化接続パラメータを使用して、パブリック証明書 *public_cert.crt* を指定します。

```
"UID=DBA;PWD=SQL;ENG=myeng;LINKS=tcpip;
 ENC=ECC_TLS (trusted_certificates=public_cert.crt)"
```

次の例では、trusted_certificates 暗号化接続パラメータを使用してパブリック証明書 *public_cert.crt* を指定し、certificate_unit、certificate_name 暗号化接続パラメータを使用して証明書フィールドを確認します。

```
"UID=DBA;PWD=SQL;ENG=myeng;LINKS=tcpip;
 ENC=ECC_TLS (trusted_certificates=public_cert.crt;

certificate_unit=test_unit;certificate_name=my_certific
ate)"
```

Web サービスでのトランスポート・レイヤ・セキュリティの使用

Web サービス用にトランスポート・レイヤ・セキュリティを設定するには、次の手順に従います。

- **デジタル証明書を作成する** パブリック証明書とサーバ証明書を作成します。パブリック証明書 (認証局の証明書の場合もあります) は、ブラウザまたは Web クライアントに配布します。サーバ証明書は、Adaptive Server Anywhere Web サーバに安全に保存されます。

  認証局の使用に関する情報を含む、デジタル証明書の一般的な情報については、「デジタル証明書の作成」37 ページを参照してください。

- **トランスポート・レイヤ・セキュリティを指定して Web サーバを起動する** -xs データベース・サーバ・オプションを使用して、HTTPS、サーバ証明書、プライベート・キーを保護するパスワードを指定します。

  次に示すのは、dbsrv9 コマンド・ラインの一部です。

  -xs *protocol***(Certificate=***server-certificate***;Certificate_Password=***password***;. . .)** . . .

  - ***protocol*** **https**、または FIPS 承認の RSA 暗号化の場合は **https_fips** を指定します。https_fips は別の承認ライブラリを使用しますが、https と互換性があります。

注意

https_fips を使用する場合は、Mozilla Firefox ブラウザに接続できます。ただし、https_fips が使用する暗号化パッケージ・プログラムは、Internet Explorer、Opera、または Safari ブラウザではサポート *されていません*。https_fips を使用する場合は、これらのブラウザに接続できません。

FIPS 承認のアルゴリズムの実行については、『ASA データベース管理ガイド』> 「-fips サーバ・オプション」を参照してください。

- ***server-certificate*** サーバ証明書のパスとファイル名を指定します。

  HTTPS では、RSA 証明書を使用する必要があります。

- ***password*** サーバ証明書のプライベート・キーのパスワードを指定します。このパスワードは、サーバ証明書を作成するときに指定します。

-xs サーバ・オプションの詳細については、『ASA データベース管理ガイド』> 「-xs サーバ・オプション」を参照してください。

Certificate パラメータと Certificate_Password パラメータの詳細については、次の項目を参照してください。

- 『ASA データベース管理ガイド』> 「Certificate プロトコル・オプション」
- 『ASA データベース管理ガイド』> 「Certificate_Password プロトコル・オプション」

- **Web クライアントの設定** ブラウザまたは他の Web クライアントがパブリック証明書を信用するように設定します。信用された証明書は、自己署名証明書、エンタープライズ・ルート証明書、または認証局証明書です。

  認証局の使用に関する情報を含む、デジタル証明書の一般的な情報については、「デジタル証明書の作成」37 ページを参照してください。

第 2 部　C2 に準拠した方法での Adaptive Server Anywhere の設定

第 2 部では、C2 に準拠した方法での Adaptive Server Anywhere の設定、インストール、実行について説明します。C2 基準を満たした構成と同様の方法で Adaptive Server Anywhere を運用するときに役立つ追加情報についても説明します。

第 3 章

インストール

この章の内容

この章では、C2 基準を満たした構成と同じ方法で Adaptive Server Anywhere (ASA) をインストールする手順について説明します。

C2 基準を満たした環境と同等の環境を保証するためには、このマニュアルの指示に正確に従ってください。

ハードウェアのインストール

『*Microsoft Windows C2 NT Administrator's and User's Security Guide*』の第 4 章に記載されている制限を考慮し、ハードウェアのユーザーズ・マニュアルの記述に従ってハードウェアを設定します。

ハードウェアの追加情報については、Sybase の Web サイトにある Final Evaluation Report (FER) を参照してください。

オペレーティング・システムのインストール

C2 基準を満たした構成を作成する最初の手順は、オペレーティング・システムのインストールと設定です。

❖ オペレーティング・システムをインストールして設定するには、次の手順に従います。

1　『Microsoft Windows NT C2 Administrator's and User's Security Guide』の第 4 章に記載されている、C2 基準を満たした構成 (Service Pack 6a と C2 セキュリティ修正プログラムを含む) に Windows NT 4.0 をインストールします。

2　管理者として Windows NT にログインします。

3　[スタート] メニューで、[プログラム] ー [管理ツール (共通)] ー [ドメイン ユーザー マネージャ] の順に選択します。

4　ユーザー マネージャを使用して sybase というユーザを作成します。

- このユーザに安全なパスワードを設定します。
- このユーザを Users グループのみに追加します。
- [ユーザーは次回ログオン時にパスワード変更が必要] チェックボックスをオフにします。
- [追加] をクリックして、[閉じる] をクリックします。

5　[原則] メニューで [ユーザーの権利] を選択します。

6　[高度なユーザー権利の表示] チェックボックスをオンにして、[権利] ドロップダウン・リストから [サービスとしてログオン] を選択します。

7　[追加] をクリックします。

ダイアログが表示されます。

8 [ドメインまたはコンピュータ] ドロップダウン・リストで ¥¥machine_name を選択します。

9 [追加する名前] フィールドに **sybase** と入力して [OK] をクリックします。

10 [OK] をクリックしてダイアログを閉じます。

11 ユーザのログオンとログオフを監査するには (これは Adaptive Server Anywhere の監査レコードを Windows ユーザに関連付けるときに役立ちます)、[原則] ー [監査] を選択してから次のように指定します。

- [監査するイベント] オプションを選択します。
- [ログオンとログオフ] の [成功] チェックボックスをオンにします。
- その他に監査するイベントがあれば選択し、[OK] をクリックします。

12 ユーザー マネージャを閉じます (省略可能)。

Adaptive Server Anywhere ソフトウェアのインストール

次に、Adaptive Server Anywhere を C2 に準拠した方法でインストールしてください。C2 に準拠するには、スタンドアロン環境で、EBF (Express Bug Fix) を適用しない英語版のみの Adaptive Server Anywhere バージョン 7.0.0 を使用してください。このマニュアルの大部分では、Adaptive Server Anywhere の現在のバージョンでの操作方法について説明していますが、この項では特に C2 基準を満たしたリリースを対象としています。

❖ Adaptive Server Anywhere 7.0.0 をインストールするには、次の手順に従います。

1　管理者として Windows NT にログインします。

2　www.sybase.com/developer から Adaptive Server Anywhere C2 パッチをダウンロードします。

3　*ASAC2Patch.exe* を実行し、ファイルをデフォルト・ディレクトリ (*C:¥ASAC2Patch*) に保存します。

ASAC2Patch.exe は自己解凍アーカイブです。

このパッチの詳細については、「Adaptive Server Anywhere の C2 パッチ」118 ページを参照してください。

4　コマンド・プロンプト・ウィンドウを開きます。

Adaptive Server Anywhere をインストールすると MDAC (Microsoft Data Access Components) も組み込まれます。MDAC のインストールによって、Windows NT システムの一部の DLL が置換されます。この DLL は、Windows NT の信頼コンピューティング基盤 (TCB: Trusted Computing Base) を構成するものです。DLL の置換を回避するには、置換対象の DLL を先にコピーしてから Adaptive Server Anywhere をインストールし、インストール後にコピーしたものを戻してください。

Adaptive Server Anywhere C2 パッチに入っている 3 つのバッチ・ファイルを使用すると、この手順を簡単に行うことができます。

最初のバッチ・ファイルを使用すると、テンポラリ・ディレクトリが作成され、そこに 14 個の *.dll* ファイルと 1 つの *.exe* ファイルが *C:¥winnt¥system32* ディレクトリからコピーされます。最初のバッチ・ファイルを実行するには、コマンド・プロンプトで次のコマンドを入力します。

```
C:
cd ¥ASAC2Patch
mdac1
exit
```

5 次の手順に従って、Adaptive Server Anywhere 7.0.0 ソフトウェアをインストールします。

- [Adaptive Server Anywhere for NetWare] チェックボックスをオフにします。
- [Adaptive Server Anywhere for Windows CE] チェックボックスをオフにします。
- [Ultra Light 開発コンポーネント] チェックボックスをオフにします。
- [同期] の下にあるすべてのオプションをオフにします。
- [PowerDynamo 3.5] オプション、[PowerDesigner] オプション、[Infomaker 7] オプションをオフにします。
- 可能であれば、[Mobile Link 同期の暗号化] チェックボックスをオフにします。
- インストール・ディレクトリはデフォルト値を使用します。

6 インストールが完了したらマシンをリブートします。

7 管理者として Windows NT にログインします。

8　*readme.txt*（*C:¥ASAC2Patch* にある) の指示に従って Adaptive Server Anywhere C2 パッチをインストールします。

この手順の後でマシンをリブートする必要はありません。

9　ソフトウェアのディレクトリに対するアクセス権を次のように設定します。

- [マイ コンピュータ] をダブルクリックします。Adaptive Server Anywhere ソフトウェアがあるディレクトリ (通常は *C:¥Program Files¥Sybase*) を右クリックして、[プロパティ] を選択します。
- [セキュリティ] タブを開き、[アクセス権] ボタンをクリックします。
- [Everyone] を選択し、[アクセス権の種類] を [読み取り] に変更します。
- [追加] をクリックします。表示されるダイアログで、[ドメインまたはコンピュータ] ドロップダウン・リストから \\machine_name を選択します。[名前] リストから [Administrators] を選択して、[追加] をクリックします。
- [ユーザの表示] をクリックします。[名前] リストから sybase を選択して [追加] をクリックします。[アクセス権の種類] を [フル コントロール] に変更して [OK] をクリックします。
- ここで説明した 3 つのエントリだけがリストに含まれるようにしてください。
- [サブディレクトリのアクセス権を置き換える] チェックボックスをオンにします。
- [OK] をクリックし、プロンプトに対して [はい] を選択します。

10 データベースとトランザクション・ログ・ファイルのフォルダを作成します。たとえば、*C:¥Databases* というフォルダを作成するとします。これ以降、このフォルダを「**C2 データベースのフォルダ**」と呼ぶことにします。このフォルダに対するアクセス権を次のように設定します。

- [マイ コンピュータ] をダブルクリックします。[Databases] フォルダを右クリックして、[プロパティ] を選択します。
- [セキュリティ] タブをクリックし、[アクセス権] ボタンをクリックします。
- [Everyone] エントリを削除します。
- [追加] をクリックします。表示されるダイアログで、[ドメインまたはコンピュータ] ドロップダウン・リストから \\machine_name を選択し、[追加する名前] フィールドに **sybase** と入力します。[アクセス権の種類] を [フル コントロール] に変更して、[OK] をクリックします。
- [OK] をクリックします。

11 エンジンがテンポラリ記憶領域として使用する *ASTMP* というフォルダを *C:¥* の下に作成します。前の手順の [Databases] フォルダと同じようにアクセス権を設定します。

12 作成したテンポラリ・フォルダに対する System 環境変数 ASTMP を設定するためには、[マイ コンピュータ] アイコンを右クリックして [プロパティ] を選択します。[環境] タブをクリックします。[システム環境変数] リストボックスで、任意のエントリをクリックします。[変数] エントリを [ASTMP] に変更し、[値] エントリを [*C:¥ASTMP*] に変更します。[設定] をクリックしてから、[OK] をクリックします。

13 Adaptive Server Anywhere C2 パッチに含まれる 2 番目のバッチ・ファイルを使用すると、*mdac1.bat* で作成されたテンポラリ・ディレクトリの *.dll* および .exe ファイルが *C:¥winnt¥system32* ディレクトリにコピーされます。2 番目の

バッチ・ファイルを実行するには、[スタート] メニューから [プログラム] ー [コマンド プロンプト] の順に選択します。コマンド・プロンプトで次のコマンドを入力します。

```
C:
 cd ¥ASAC2Patch
 mdac2
 exit
```

14 Windows NT を C2 基準を満たした構成にすると、いくつかのレジストリ・キーが削除されます。Adaptive Server Anywhere のインストール時に、このうち 2 つのキーが再作成されます。Windows NT を C2 基準を満たした構成にしておくには、この 2 つのキーを再び削除してください。*regedt32.exe* を使用すると、次のレジストリ・キーが削除されます。

キー	**HKEY_LOCAL_MACHINE¥SOFTWARE**
サブキー	Microsoft¥OS/2 Subsystem for Windows NT
エントリ	delete all subkeys

キー	**HKEY_LOCAL_MACHINE¥SYSTEM**
サブキー	CurrentControlSet¥Control¥Session Manager¥Environment
エントリ	Os2LibPath
値	delete entry

15 また、これらのファイルには次のような正しいアクセス権があることを確認してください。

ファイル	**C2 レベルのアクセス権**
BOOT.INI、NTDETECT.COM、NTLDR	Administrator: フル コントロール SYSTEM：フル コントロール

16 開いているウィンドウをすべて閉じて、マシンをリブートします。

マシンをリブートするのは、サービス コントロール マネージャにシステム環境変数の変更を読み込ませるためです。

17 管理者として Windows NT にログインします。

18 Adaptive Server Anywhere C2 パッチに含まれている 3 番目のバッチ・ファイルを使用すると、*mdac1.bat* で作成されたテンポラリ・ディレクトリがクリーンアップされます。3 番目のバッチ・ファイルを実行するには、コマンド・プロンプト・ウィンドウを開きます。コマンド・プロンプトで次のコマンドを入力します。

```
C:
 cd ¥ASAC2Patch
 mdac3
 exit
```

データベースの作成

C2 準拠の構成で運用するためには、データベースも C2 準拠である必要があります。データベースに接続するときは必ず統合化ログイン・メカニズムを使用してください。データベースへの標準接続 (たとえば、ユーザ ID とパスワードを指定した接続) は C2 基準を満たした構成では認められていません。

❖ C2 準拠データベースを作成するには、次の手順に従います。

1　sybase としてログインします。

2　[スタート] メニューから、[プログラム] ー [コマンド プロンプト] の順に選択します。

3　次の制限に従って dbinit ユーティリティを使用し、データベースを作成します。

- `-I` スイッチを使用して、jConnect サポートを無効にしてください。
- `-k` スイッチまたは `-n` スイッチは使用しないでください。
- 作成したデータベース・ファイルを C2 データベースのフォルダに入れてください。
- `-t` スイッチを使用してトランザクション・ログ・ファイルを指定する場合、または `-m` スイッチを使用してトランザクション・ログ・ミラー・ファイルを指定する場合は、C2 データベースのフォルダに入れてください。

  C2 基準を満たした構成での dbinit ユーティリティの使用方法の詳細については、「初期化ユーティリティ」107 ページを参照してください。データベースのフォルダの詳細については、「Adaptive Server Anywhere ソフトウェアのインストール」61 ページを参照してください。

4 データベースを作成したら、データベースに接続する必要があります。

この接続では、min_password_length オプションと DBA のパスワードの設定のみを行います。

5 コマンド・プロンプトで、**dbisqlc −c UID=DBA;PWD=SQL;DBF=file** と入力します。このとき、*file* にはさきほど作成したデータベース・ファイルのフル・パスを指定します。

数秒で Interactive SQL が表示されます。

C2 基準を満たした構成での dbisqlc ユーティリティの使用方法の詳細については、「Interactive SQL ユーティリティ」111 ページおよび「制限」86 ページを参照してください。

6 set option public.min_password_length=6 (または 6 よりも大きな値) と入力し、[実行] をクリックします。

7 grant connect to DBA identified **by newpw** と入力します。このとき、*newpw* には DBA アカウントの新しいパスワードを指定します。[実行] をクリックします。

新しいパスワードは手順 5 で入力した文字数以上の長さにし、推測しやすい文字列は使用しないでください。

8 grant integrated login to sybase as user DBA と入力し、[実行] をクリックします。

9 set option public.login_mode='Integrated' と入力し、[実行] をクリックします。

10 ウィンドウの右上隅の [X] をクリックして Interactive SQL を終了します。

データベース・エンジンの実行

1. 管理者として Windows NT にログインします。

   サービスの作成、開始、停止を行うには、管理者権限が必要です。

2. コマンド・プロンプトを開きます。

3. 次の制限に従って dbsvc ユーティリティを使用し、サービスを作成します。

   - -a スイッチを使用して sybase アカウントを指定し、-p スイッチを使用してパスワードを指定します。
   - -as スイッチまたは -i スイッチは使用しないでください。
   - 実行プログラムの名前は次のように指定してください。

     ```
     C:¥Program Files¥Sybase¥SQL Anywhere 9¥
     win32¥dbeng9.exe
     ```

     これはパーソナル・データベース・サーバの場合です。

     ```
     C:¥Program Files¥Sybase¥SQL Anywhere 9¥
     win32¥dbsrv9.exe
     ```

     これはデータベース・サーバの場合です。

   - 次のエンジン・パラメータを使用します。

     - -n *engine name*
     - -sc
     - -gd DBA
     - -gk DBA
     - -gl DBA
     - -gu DBA
     - -x namedpipes(TDS=NO)

4. 実行するデータベース・ファイル名は必ずフル・パスで指定します。

   パスは *database-folder¥filename.db* の形式で指定します。このとき、*database-folder* には C2 データベースのフォルダを指定し、その他関連するパラメータがあれば指定します。

   たとえば、以下のコマンド・ラインのように指定すると、手動で開始し、ネットワーク・サーバを参照する asa_svc と呼ばれるサービスが作成されます。このサービスは、パスワードが sybase_password の sybase アカウントで実行します。これは、次のコマンドを実行します。

   ```
   C:¥Program Files¥Sybase¥SQL Anywhere 9¥win32¥
    dbsrv9.exe -n asa_c2 -sc -gd DBA -gk DBA
    -gl DBA -gu DBA -x namedpipes(TDS=NO)
    database-folder¥c2test.db
    dbsvc -a sybase -p sybase_password -s manual
    -t network -w asa_svc C:¥Program Files¥Sybase¥
    SQL Anywhere 9¥win32¥dbsrv9.exe -n asa_c2 -sc
    -gd DBA -gk DBA -gl DBA -gu DBA
    -x namedpipes(TDS=NO) database-folder¥c2test.db
   ```

   C2 基準を満たした構成でのエンジンとサーバの使用方法の詳細については、「データベース・エンジン／サーバ」103 ページを参照してください。

5. サービスを開始および停止するには、[コントロール パネル] で Windows NT サービス・マネージャを実行します。[スタート] メニューから [設定] ー [コントロール パネル] の順に選択し、[サービス] をダブルクリックします。

   ここで作成したサービスが Adaptive Server Anywhere - *svc* の下に表示されます。このとき *svc* は、dbsvc コマンド・ラインで指定したサービス名です。

6. [開始] ボタンと [停止] ボタンを使用して、サービスの開始と停止を行います。

第 4 章

監査

この章の内容

この章では、監査出力の読み込み方法や Adaptive Server Anywhere 監査出力と Windows NT 監査を関連付ける方法について説明します。

監査の有効化と無効化

データベースを作成するとき監査は OFF になります。ただし、監査の public オプションを使用するといつでも監査の有効と無効を切り替えられます。

❖ 特定のデータベースの監査を開始するには、次の手順に従います。

- 次の SQL 文を使用してオプションを ON にします。

```
SET OPTION public.auditing='on'
```

public オプションを設定できるのは DBA 権限を持つユーザのみです。一度このオプションを有効にすると、すべてのパーミッション・チェックと接続試行が監査されます。

❖ 特定のデータベースの監査を停止 (無効化) するには、次の手順に従います。

- 次の SQL 文を使用してオプションを OFF にします。

```
SET OPTION public.auditing = 'off'
```

この文を発行できるのは、DBA 権限を持つユーザのみです。

エンジンまたはサーバによって生成される監査レコードの各タイプの詳細と完全なリストについては、「監査レコード」75 ページを参照してください。

注意 C2 基準を満たした構成で実行する場合、監査はオプションです。

監査出力の読み込み

dbtran ユーティリティを使用して、トランザクション・ログから監査レコードを取り出すことができます。トランザクション・ログは、通常、データベース・ファイルと同じディレクトリの *dbname.log* ファイルにあります。

dbtran に -g スイッチを指定すると、監査レコードが出力に含まれます。dbtran からの出力は、コメントが混在する SQL スクリプトです。この SQL スクリプトは、障害が発生したときにデータベースをリカバリするために使用できます。-g オプションを使用すると出力ファイル全体がコメントになります。これは、-g オプションによって -d オプションが暗黙に指定されるためです (このとき、トランザクション・ログ情報は、デフォルトのコミット順ではなくログに含まれる順序で記録されます)。このフォーマットの出力は、データベースのリカバリに使用しないでください。誤ってこのファイルをリカバリに使用しないように、すべての行がコメントになっています。

ユーザがデータベースに接続すると、次の監査レコードが生成されます。

```
--CONNECT-1001-0000198970-dba-1998/dec/03 14:54
```

CONNECT の後のデータは次のように解釈されます。

- 1001 はこの接続に割り当てられた接続 ID です。これ以降、次に CONNECT-1001 が出現するまでの、接続 ID 1001 を含むすべてのトランザクションは、この接続に属します。
- 0000198970 は、トランザクション・ログ内にあるレコードのバイト・オフセットです。
- dba は、この接続にログインしたユーザ名です。
- 1998/dec/03 14:54 は接続の日付と時刻です。

他のレコードには接続 ID とバイト・オフセットがありますが、ユーザ名と日付／時刻を含むのは CONNECT レコードのみです。切断については記録されないことに注意してください。前の CONNECT レコードと同じ接続 ID を持つ別の CONNECT レコードが生成された場

合は、最初の接続は切断していると考えられます。この接続 ID が再び使用されても、2 番目の接続は最初の接続とはまったく関係ありません。

監査レコード

この項では、エンジンまたはサーバによって生成される可能性のあるさまざまな監査レコード、レコードに含まれる情報、レコードが生成される状況について説明します。dblog、dbtran、dbwrite の 3 つのデータベース・ユーティリティによって *.alg* ファイルに生成される監査レコードの説明については、「データベース・ユーティリティの監査」83 ページを参照してください。

タイプ	情報	使用方法
オペレーションを試みています	日付／時刻、試行された操作の SQL	このレコードには、試行中の操作が表示されます。これは、トランザクション・ログが機能するために必要です。 トランザクション・ログには、リカバリが必要になった場合にデータベースのデータまたはスキーマに対する変更内容をレプリケートするための SQL が含まれます。監査レコードがこのログに組み込まれると、パーミッション・チェックが発生するごとに記録され、データベースでのアクティビティを後で再生できます。 ただし、パーミッション・チェックが失敗すると、試行中の操作は実際には行われないため、記録もされません。この場合、試行内容を判別する方法はありません。これが特に重要になるのは、DBA 以外のユーザが、DBA 権限を必要とする操作を試行する場合です。 このため、すべての DDL 文 (および一部のその他の文) は試行前に記録されます。

タイプ	情報	使用方法
オペレーションが成功しました／失敗しました	日付／時刻、成功または失敗	このレコードは、同一の接続 ID について、最後の Operation Attempt レコード、Attempting to set public option レコード、または Attempting SETUSER レコードが成功と失敗のどちらであるかを示します。
パーミッションを確認しています	日付／時刻、パーミッション／権限の種類、テーブル名 (該当する場合)、カラム名 (該当する場合)、プロシージャ／関数名 (該当する場合)	このレコードは、特定のパーミッションまたは権限のチェックが行われたことを示します。該当するパーミッションは次のいずれかです。 DBA / Resource 権限 テーブルに対する Insert / Update / Select / Delete / Alter / Resource パーミッション テーブルとカラムに対する Update / Select / Resource パーミッション テーブルに対する Grant Insert / Update / Select / Delete / Alter / Resource パーミッション テーブルとカラムに対する Grant Update / Select / Resource パーミッション プロシージャまたは関数に対する Execute パーミッション プロシージャまたは関数に対する Grant Execute パーミッション

タイプ	情報	使用方法
ユーザを確認しています	日付／時刻、ユーザ名	このレコードは、ユーザのチェックが行われたことを示します。これは、オブジェクトの所有権を判別するときに役立ちます。たとえば、ユーザ bob がテーブル T を所有しているとします。テーブル T への挿入が試行された場合に、現在のユーザがユーザ bob かどうかを確認します。レコードのテキストの「確認しています」は、現在のユーザ名を確認していることを示しています。
パブリック・オプションを設定しようとしています	日付／時刻、オプション名	このレコードは、PUBLIC ユーザが所有するオプションをあるユーザが設定しようとしたことを示します。このオプションを実行できるのは DBA 権限を持つユーザのみです。このため、この試行では必ず DBA 権限のチェックが行われます。Operation Succeeded/Failed レコードに成功または失敗が示されます。
監査が有効です／無効です	日付／時刻	このレコードは、監査の public オプションが変更されたことを示し、必ず Set Public Option レコードの次にあります。また、監査が有効であるか無効であるかに関係なく生成されます。ただし、監査がすでに有効になっているときに監査変数を ON に設定したり、監査がすでに無効になっているときに変数を OFF に設定したりした場合には、このレコードは生成されません。

タイプ	情報	使用方法
SETUSER を試みています	日付／時刻、ユーザ名	このレコードは、ユーザがパラメータを指定して SETUSER コマンドを試行したことを示します。これを実行できるのは DBA 権限を持つユーザのみです。このため、この試行では必ず DBA 権限のチェックが行われます。Operation Succeeded/Failed レコードに成功または失敗が示されます。引数なしの SETUSER コマンドは、すべてのユーザが実行できるため、監査も記録も行われないことに注意してください。
接続の試行	日付／時刻、ユーザ名 (成功した場合)、マシン・アドレス (同一マシンの場合はローカル)、ポート・タイプ、成功または失敗	このレコードは、接続の試行が行われたことを示します。
トリガが起動されました／終了しました	日付／時刻、トリガ名	このレコードは、トリガが起動または終了したことを示します。この 2 つのレコードの間の同一接続に対するすべての監査レコードでは、トリガの実行が監査されます。トリガは、呼び出し元ではなくテーブル所有者のパーミッションを使用して実行するため、Trigger firing レコードと Trigger finishing レコードの間で監査されるすべてのパーミッション・チェックはテーブル所有者に関して実行されることに注意してください。トリガの起動を引き起こした SQL 文を調べると、テーブル所有者がわかります。Trigger firing レコードの直前の同一接続に対する SQL 文を調べてください。この SQL 文は、テーブルに対する挿入、更新、または削除のいずれかです。テーブル名は、owner.table というフォーマットで示されます。

タイプ	情報	使用方法
String	日付／時刻、文字列	このタイプのレコードは、sa_audit_string と呼ばれるシステム・ストアド・プロシージャを使用して監査証跡に挿入できます。このプロシージャを実行できるのは DBA 権限を持つユーザのみです。任意の文字列 (最長 128 文字) を指定できます。

表 6.2 - 監査レコードのフォーマット - 固定

タイプ	フォーマット
トランザクション再実行ヘッダ	1 バイト
接続識別子	3 バイト
日付／時刻	11 バイト • 2 バイト：年 (例、1998) • 1 バイト：月 (1 ～ 12) • 1 バイト：日 (1 ～ 31) • 1 バイト：時 (0 ～ 23) • 1 バイト：分 (0 ～ 59) • 1 バイト：秒 (0 ～ 59) • 4 バイト：マイクロ秒 (0 ～ 999999)
監査タイプ	1 バイト

表 6.3 - 監査レコードのフォーマット - タイプ別変数

タイプ	フォーマット
AUDIT_ENABLE_AUDITING	• 1 バイト: 1 (監査が有効) または 0 (監査が無効)
AUDIT_SET_PUB_OPTION	• 2 バイト：後続の文字列の長さ (n) • n バイト：オプション名
AUDIT_OP_ATTEMPT	• 2 バイト：後続の文字列の長さ (n) • n バイト：試行する操作の SQL
AUDIT_OP_SUCCESS	• 1 バイト: 1 (操作成功) または 0 (操作失敗)
AUDIT_PERM_CHECK	• 1 バイト : 1 (成功) または 0 (失敗) • 2 バイト：後続の文字列の長さ (n) • n バイト : パーミッション・タイプ (たとえば、select、update、execute) • 2 バイト：後続の文字列の長さ (n) • n バイト : オブジェクト (テーブル、ビュー、プロシージャなど) の名前 • 2 バイト：後続の文字列の長さ (n) • n バイト : カラム名 (該当する場合)
AUDIT_USER_CHECK	• 1 バイト : 1 (成功) または 0 (失敗) • 2 バイト：後続の文字列の長さ (n) • n バイト：ユーザ名
AUDIT_CONNECTION	• 1 バイト : 1 (成功) または 0 (失敗) • 2 バイト：後続の文字列の長さ (n) • n バイト：ユーザ名 (接続が成功した場合) • 2 バイト：後続の文字列の長さ (n) • n バイト：マシン ID
AUDIT_SETUSER	• 2 バイト：後続の文字列の長さ (n) • n バイト：ユーザ名

タイプ	フォーマット
AUDIT_TRIGGER	• 2 バイト：後続の文字列の長さ (n) • n バイト：トリガ名 • 3 バイト：起動または終了
AUDIT_STRING	• 2 バイト：後続の文字列の長さ (n) • n バイト：変数テキスト文字列

監査レコードの管理

ログ変換 [dbtran] ユーティリティを使用すると、トランザクション・ログから監査レコードを取り出すことができます。dbtran を呼び出すときに -u スイッチまたは -x スイッチを使用すると、ユーザ名に応じてレコードをフィルタできます。監査レコードは削除できません。ただし、dblog ユーティリティまたは dbbackup ユーティリティを使用すると、トランザクション・ログをパージまたはトランケートできます。

トランザクション・ログのパージの詳細については、「トランザクション・ログ・ユーティリティ」110 ページを参照してください。

監査ログ (Adaptive Server Anywhere の場合はトランザクション・ログ) がいっぱいになると、エンジンまたはサーバは保留中のすべてのトランザクションをロールバックし、以降の要求はすべて失敗します。引き続きデータベースを使用するには、この時点でトランザクション・ログをトランケートしてください。トランザクション・ログをバックアップしてからトランケートすることを強く推奨します。エンジンを停止してから、ファイルを別のディスクにコピーするのが、トランザクション・ログをバックアップする最も簡単な方法です。その後、古いトランザクション・ログ・ファイルを削除し、エンジンまたはサーバを再起動します。新しいトランザクション・ログ・ファイルが作成されます。

データベース・ユーティリティの監査

一部のデータベース・ユーティリティは、監査が必要な動作を実行しますが、必ずしも実行中のエンジンまたはサーバと通信するとは限りません。これらのユーティリティは個別に監査してください。該当するユーティリティは dblog、dbwrite、dbtran です。これらのユーティリティは、データベースまたはトランザクション・ログをチェックして、監査が有効になっているかどうかを調べます。監査が有効である場合、これらのユーティリティは、データベース・ファイルと同じディレクトリにある *dbname.alg* というファイルに書き込むことによって、自らの呼び出しを監査します。

.alg ファイルはテキスト・ファイルであり、メモ帳などの標準エディタで表示できます。また、テキスト・ファイルのソートやフィルタのユーティリティ (grep など) を使用して、特定のユーザまたはユーティリティに関する監査レコードを取り出すこともできます。

各監査レコードは 1 つの行で構成され、フォーマットは次のようになります。

```
2000/07/07 15:31:17.316 - User NT user name invoking
utility name
```

エディタでレコードを削除しファイルを保存することによって、いつでもこのファイルのレコードを削除できます。ファイル自体もいつでも削除できます。このファイルに書き込めないと (たとえば、ファイル・システムがいっぱいの場合)、このファイルにレコードを生成するユーティリティは失敗します。Windows NT の監査メカニズムを使用して、*.alg* ファイルへのアクセスを監査できます。

監査レコードの関連付け

ある監査レコードが生成されたときに Windows NT にログインしていたユーザの名前がわかると役に立つ場合があります。たとえば、ログオンの試行が集中して何度も失敗していることに DBA が気づいたときに、Windows NT にログインしていたユーザを知りたい場合があります。必要な情報の種類によって異なりますが、これには 2 つの方法があります。

この例では、問題の監査イベントが発生した時刻を記録するだけです。すべての監査イベントにはイベントの日付と時刻が含まれています。次に、管理者として Windows NT にログインし、イベント・ビューア・アプリケーションを実行します。[ログ] メニューから [セキュリティ] を選択して、ログオンとログアウトの監査レコードを確認します。問題の監査イベントの日付と時刻の直前のログオン／ログオフ・イベントを見つけてダブル・クリックします。これはログオンの成功イベントです。ログオンしたユーザ名とユーザのドメインが表示され、監査されたイベントが発生したときに Windows NT ワークステーションにログインしていたユーザがわかります。これが可能となるのは、Windows NT のログオンとログアウトの監査が「オペレーティング・システムのインストール」59 ページ中に有効になっていた場合のみです。

次に示す 2 つ目は、監査ログに特定の接続に関する情報が含まれており、それを特定の Windows ユーザと関連付ける必要がある場合に便利です。すべての接続に統合化ログインが使用されるので、データベース・ユーザは特定の Windows ユーザにマップされます。このマッピングは常に 1 対 1 であるため、該当するデータベース・ユーザにマップできる Windows ユーザは他にいないことがわかります。データベース・ログイン ID を与えられた Windows ユーザの名前を調べるには、次の SQL 文を実行してください。

```
SELECT lg.integrated_login_id
 FROM syslogin lg
 KEY JOIN sysuserperm p
 WHERE p.user_name='login ID'
```

第 5 章

制限とその他のセキュリティの考慮事項

この章の内容　この章では、C2 基準の制限とその他のセキュリティの考慮事項について説明します。

制限

Adaptive Server Anywhere を C2 基準を満たした構成で実行するには、次の制限が必要です。

1. Adaptive Server Anywhere インストール・ディレクトリのファイルを削除、修正、または置換しないでください。ただし、次のファイルは例外です。

   - *win32¥util_db.ini* - このファイルは必要に応じて修正できます。
   - *win32¥asasrv.ini* - このファイルは必要に応じて修正または削除できます。
   - *win32¥rebuild.bat* - このファイルは必要に応じて修正できます。
   - *win32¥backup.syb* - このファイルは必要に応じて修正または削除できます。
   - *win32¥procdebug.bat* - このファイルは必要に応じて修正できます。
   - *win32¥custom.SQL* - このファイルは必要に応じて修正できます。
   - *win32¥tjava.pdf* - このファイルは必要に応じて削除できます。

2. Adaptive Server Anywhere インストール・ディレクトリに新規ファイルを追加しないでください。

3. sybase アカウントのパスワードを与えられるユーザは 1 名だけです。

4. sybase アカウントのパスには、*%SystemRoot%¥system32*、*%SystemRoot%*、Adaptive Server Anywhere *win32* ディレクトリ以外のディレクトリは指定できません。

5. sybase アカウントにはサービスとしてログインする権限のみを付与してください。

6. DBA 権限は非常に強力です。必要とするユーザにのみ DBA 権限を付与します。DBA ユーザ数は最小限に保ってください。DBA 権限を必要とする各ユーザには個別のアカウントを与えて、そのアカウントに DBA 権限を付与してください (共有 DBA アカウントは使用しないでください)。

7. DBA 権限の範囲に含まれないデータベースを使用する DBA には、2 つの異なる Adaptive Server Anywhere ユーザ・アカウント (DBA 権限を含むアカウントと含まないアカウント) を与えてください。DBA は、必要な場合にかぎり DBA 権限付きのアカウントを使用してください。

8. 新規データベースの作成時には DBA アカウントのパスワードを変更してください。

9. 新規データベースの作成時には、min_password_length public オプションの値を少なくとも 6 に設定してください。

10. データベース・エンジンまたはサーバを Windows NT サービスとして実行してください。Adaptive Server Anywhere は、サービスとして実行する場合にのみ動作確認されています。

11. エンジンまたはサーバの startline に次のスイッチを指定してください。

    ```
    -sc -gd DBA -gk DBA -gl DBA -gu DBA
     -x namedpipes(TDS=NO)
    ```

    エンジンまたはサーバの startline を指定するのは dbsvc ユーティリティを実行するときです。このため、これらのスイッチを dbsvc コマンドの Details 部に組み込む必要があります。

    詳細については、dbsvc の詳細に関する「サービス作成ユーティリティ」108 ページを参照してください。

12. 名前付きパイプ以外のポートを開始するときは -x パラメータを使用しないでください。Adaptive Server Anywhere の動作が確認されているのはスタンドアロン環境でのみです。

13. REMOTE_DBA 権限はどのユーザにも付与しないでください。

14. 次のシステム・プロシージャの execute パーミッションはどのユーザまたはグループにも付与しないでください。

    - xp_cmdshell
    - xp_startmail
    - xp_sendmail
    - xp_stopmail
    - xp_read_file
    - xp_write_file
    - sp_audit_string
    - java_debug_version
    - java_debug_connect
    - java_debug_disconnect
    - java_debug_get_existing_vms
    - java_debug_free_existing_vms
    - java_debug_wait_for_debuggable_vm
    - java_debug_get_vm_name
    - java_debug_release_vm
    - java_debug_attach_to_vm
    - java_debug_detach_from_vm
    - java_debug_detach_request
    - バージョン 7 よりも後に導入されたすべてのシステム・プロシージャ。

15. DBA 権限を持つユーザが所有するストアド・プロシージャまたは関数を作成しないでください。

16. DBA 権限を持つユーザが所有するテーブルにはトリガを作成しないでください。

17. 古いデータベースを使用するときは、前もって dbupgrad ユーティリティを実行してアップグレードしてください。

    データベースのアップグレードの詳細については、『ASA データベース管理ガイド』>「dbupgrad コマンド・ライン・ユーティリティを使用したデータベースのアップグレード」を参照してください。

18. データベースにはトランザクション・ログ・ファイルを必ず使用してください。データベースを作成するときは -n スイッチ (トランザクション・ログなし) を使用しないでください。また、データベースに対して dblog -n (トランザクション・ログまたはミラーを使用しない) を実行しないでください。

19. すべてのデータベース、トランザクション・ログ、dbspace、ライト・ファイル、ミラー・ファイルは、共有しない保護ディレクトリに格納してください。

    ディレクトリ保護のガイドラインについては、「Adaptive Server Anywhere ソフトウェアのインストール」61 ページを参照してください。

20. java.net パッケージはエンジンまたはサーバでは無効です。データベースで実行する Java では、このパッケージは使用できません。

21. java_input_output public オプションは常に OFF (デフォルト) に設定してください。

22. guest というデータベース・ユーザは作成しないでください。このようなユーザを作成すると、すべての Windows ユーザが統合化ログインを使用してデータベースに接続できてしまいます。

23. データベースのインストール時には login_mode public オプションを常に Integrated に設定してください。

    詳細については、「データベースの作成」67 ページを参照してください。

24. データベースに接続するときは必ず統合化ログイン・メカニズムを使用してください。データベースへの標準接続 (ユーザ ID とパスワードを指定する接続) は C2 基準を満たした構成では認められていません。

25. 統合化ログイン・マッピングを必ず 1 対 1 にしてください。2 つの Windows ユーザ名が同一データベース・ユーザにマップされないようにしてください。

26. Embedded SQL プログラムでは db_delete_file 関数を使用しないでください。削除されるファイルの名前は監査されないためです。

27. *sys.sysuserperm* または *sys.syslogin* に対する SELECT アクセスを非 DBA ユーザに付与しないでください。

セキュリティの警告

次に、注意する必要があるその他のセキュリティ上の問題を示します。

1. トリガはテーブル所有者のパーミッションを使用して実行するため、あるテーブルに対する ALTER パーミッションを持つすべてのユーザは、自分が所有する他のテーブルにアクセスするトリガを作成できます。1 つのテーブルに対する ALTER パーミッションを別のユーザに付与すると、自分が所有するすべてのテーブルに対するすべてのパーミッションをそのユーザに付与することになるので注意してください。

2. トリガが起動されたとき、およびトリガによって実行されたストアド・プロシージャが終了したときに監査レコードが作成されます。これらの監査レコードに示されるユーザ ID は、トリガが定義されているテーブルの所有者の ID です。

3. ストアド・プロシージャには GRANT コマンドを組み込むことができます。そのようなプロシージャを実行する場合、呼び出し元のパーミッションではなくストアド・プロシージャの所有者のパーミッションを使用して GRANT が実行されます。GRANT 文を組み込んだストアド・プロシージャを作成する場合は、上記のことに注意してください。

4. Windows NT には、ユーザが実行するアクションを監査する機能があります。sybase ユーザを監査するように、ユーザ自身で Windows NT を構成することをおすすめします。このような監査によってデータが大量に生成される場合があるので注意してください。

   詳細については、「オペレーティング・システムのインストール」59 ページを参照してください。

5. テーブルとカラムのパーミッションは累積されますが、それぞれ独立しています。つまり、2 つの異なる GRANT 文を実行すると、パーミッションが重複する場合、2 つのうち 1 つを取り消しても残りの 1 つは取り消されません。

たとえば、ユーザ fred が sue に対して Employee テーブルの GRANT UPDATE (Street) を実行すると、Sue は Employee テーブルの Street カラムを更新できます。

さらに、ユーザ fred が sue に対して Employee テーブルの GRANT UPDATE を実行すると、Sue は Employee テーブルのすべてのカラムを更新できるようになります。

次に、ユーザ fred が sue に対して Employee テーブルの REVOKE UPDATE を実行すると、2 つ目の付与は取り消されますが最初の付与は依然として有効です。Sue は引き続き、テーブル Employee の Street カラムを更新する権限を持ちます。

ネストされたオブジェクトの所有権の変更

ビューとプロシージャは、さまざまなユーザが所有する基本のオブジェクトにアクセスできます。たとえば、usera、userb、userc、userd が別々の 4 名のユーザである場合は、userc.viewc から userd.viewd を作成できます。この userc.viewc は、usera.table から作成された userb.viewb をベースにして作成できます。同じように、プロシージャでも、userd.procd は userc.procc を呼び出すことができ、userc.procc は、usera.tablea に挿入できる userb.procb を呼び出すことができます。

ネストされたビューおよびテーブルには、次の DAC (任意アクセス制御) 規則が適用されます。

- ビューを作成するには、ユーザは、そのビューのすべてのベース・オブジェクト (たとえばテーブルとビュー) の SELECT パーミッションが必要です。
- ビューにアクセスするには、ビューの所有者は、基本のテーブルまたはビューに対する適切なパーミッションを GRANT オプションで付与されている必要があります。また、ユーザは、ビューに対する適切なパーミッションを付与されている必要があります。
- WHERE 句を使用して更新するには、SELECT パーミッションと UPDATE パーミッションの両方が必要です。
- ユーザがビュー定義内のテーブルを所有している場合は、そのユーザがビューの所有者ではなく、ビューに対するアクセス権を付与されていなくても、ビューを介してテーブルにアクセスできます。

ネストされたプロシージャには次の DAC 規則が適用されます。

- プロシージャを作成する場合、ユーザは、基本となるオブジェクト (たとえば、テーブル、ビュー、またはプロシージャ) に対するパーミッションを必要としません。
- プロシージャを実行する場合、プロシージャの所有者は、そのプロシージャが参照するオブジェクトに対する適切なパーミッションを必要とします。

- プロシージャによって参照されるすべてのテーブルをユーザが所有する場合でも、プロシージャに対する EXECUTE パーミッションを付与されていないかぎり、プロシージャを実行してテーブルにアクセスすることはできません。

この動作については次の例で説明します。

例 1: user1 が table1 を作成し、user2 が table1 について view2 を作成する

- user1 は所有者であるため、常に table1 にアクセスできます。
- user1 は、基本となるテーブルの所有者であるため、常に view2 を介して table1 にアクセスできます。これは、user2 が view2 のパーミッションを user1 に付与しない場合でも該当します。
- user2 が table1 に直接または view2 を介してアクセスできるのは、user1 が table1 のパーミッションを user2 に付与した場合です。
- user3 が table1 にアクセスできるのは、user1 が table1 のパーミッションを user3 に付与した場合です。
- user3 が view2 を介して table1 にアクセスできるのは、user1 が table1 のパーミッションを user2 に grant オプションとともに付与し、user2 が view2 のパーミッションを user3 に付与した場合です。

例 2: table1 にアクセスする procedure2 を user2 が作成する

- user1 が procedure2 を介して table1 にアクセスできるのは、user2 が procedure2 の EXECUTE パーミッションを user1 に付与した場合です。view2 では、user1 はパーミッションを必要としませんが、上記の場合とは異なるので注意してください。

例 3: user1 が table1 を作成し、user2 が table2 を作成し、user3 が table1 と table2 をジョインする view3 を作成する

- user3 が view3 を介して table1 と table2 にアクセスできるのは、user1 が table1 のパーミッションを user3 に付与し、user2 が table2 のパーミッションを user3 に付与した場合です。
- user3 が table1 のパーミッションを持っているが table2 のパーミッションを持っていない場合、user3 は view3 を使用できません。table1 のカラムからなるサブセットにもアクセスできません。
- user1 または user2 が view3 を使用できるのは、(a) user1 が table1 のパーミッションを grant オプションで user3 に付与し、(b) user2 が table2 のパーミッションを grant オプションで user3 に付与し、(c) user3 が view3 のパーミッションを user1 または user2 に付与した場合です。

DBA 権限の取り消し

通常、エンジンでは、ユーザがデータベースに接続している間は DBA 権限を取り消すことはできません。このため、DBA 権限を取り消す最も簡単な方法は、ユーザが切断するまで待ってから REVOKE DBA 文を発行することです。

ただし、場合によっては、現在データベースに接続しているユーザが何も実行しないうちに、ただちに DBA 権限を取り消す必要があります。ユーザ fred の DBA 権限を取り消す例について考えてみます。

❖ 接続しているユーザの DBA 権限を取り消すには、次の手順に従います。

1 DBA 権限を持つ*別の*ユーザとして同じデータベースに接続します。

つまり、fred 以外のユーザ ID を使用します。

2 次の文を実行して、サーバへの接続を無効にします。

```
CALL sa_server_option('ConnsDisabled', 'ON')
```

これで、fred の既存の接続を切断すると fred は再接続できなくなります。

3 次の文を実行して、データベースのすべての接続をリストします。

```
CALL sa_conn_info( )
```

4 Userid カラムが fred であるローごとの Number カラムの値を書き留めます。

5 手順 4 で書き留めた接続番号ごとに、次の文を実行します。

```
DROP CONNECTION number
```

これで、各接続がただちに切断され、コミットされていないトランザクションはロールバックされます。fred によってコミットされたトランザクション、および DROP 文の実行前に fred が実行した DDL はロールバックされないことに注意してください。手動で元に戻してください。

6　次の SQL 文を実行します。

```
REVOKE DBA FROM fred
```

7　次の文を実行して、サーバへの接続を再び有効にします。

```
CALL sa_server_option('disable_connections', 'OFF')
```

TCB サブセット

C2 基準を満たした構成に含まれる信頼コンピューティング基盤 (TCB: Trusted Computing Base) を構成するソフトウェア・モジュールおよびファイルを次に示します (すべての *.exe* ファイルと *.dll* ファイルは Adaptive Server Anywhere ディレクトリの *win32* サブディレクトリにあります)。

1. データベース・エンジン／サーバ
    - *dbeng9.exe*
    - *dbsrv9.exe*
    - *dbserv9.dll*
    - *dbctrs9.dll*
    - *libsybbr.dll*
    - *dblgen9.dll*
    - *dbcis9.dll*
    - *dbjava9.dll*
    - **scripts* ディレクトリの **.sql*
    - *java* ディレクトリの **.zip*
2. Interactive SQL
    - *dbisqlc.exe*
    - *dbcon9.dll*
    - *dblgen9.dll*
    - *dbtool9.dll*
    - *dblib9.dll*
3. データベース・ユーティリティ
    - *dbackup.exe*

- *dbcollat.exe*
- *dbdsn.exe*
- *dberase.exe*
- *dbexpand.exe*
- *dbinfo.exe*
- *dbinit.exe*
- *dblog.exe*
- *dbping.exe*
- *dbshrink.exe*
- *dbstop.exe*
- *dbsvc.exe*
- *dbtran.exe*
- *dbunload.exe*
- *dbupgrad.exe*
- *dbvalid.exe*
- *dbwrite.exe*
- *sqlpp.exe*
- *dblgen9.dll*
- *dbtool9.dll*
- *dblib9.dll*

第 6 章

制限付き構文

この章の内容

この章では、エンジンとサーバの構文と、C2 基準を満たした構成で使用されるいくつかのデータベース・ユーティリティについて説明します。

制限付き構文

この項では、エンジンとサーバの構文と、C2 基準を満たした構成で使用されるいくつかのデータベース・ユーティリティについて説明します。これらのツールについては『ASA データベース管理ガイド』>「データベース管理ユーティリティ」に詳細がありますが、ここでも参照しやすいように説明します。また、特に C2 基準を満たした構成での必須スイッチまたは制限付きスイッチについて説明します。オプションのスイッチがリストされている場合、使用できるのはリストされているスイッチのみであることに注意してください。ユーティリティの使用画面で説明またはリストされていても、ここにリストされていないスイッチは C2 基準を満たした構成では使用できません。

各スイッチの詳細については、『Adaptive Server Anywhere リファレンス・マニュアル』を参照してください。

データベース・エンジン／サーバ

構文 1　**dbeng9 -sc -gd dba -gk dba -gl dba -gu dba -x namedpipes(TDS=NO)**
[*optional-engine-or-server-switches*]
[*db-file* [*optional-database-switches*]] …

構文 2　**dbsrv9 -sc -gd dba -gk dba -gl dba -gu dba -x namedpipes(TDS=NO)**
[*optional-engine-or-server-switches*]
[*db-file* [*optional-database-switches*]] …

必須スイッチ：

スイッチ	説明	理由
-sc	C2 基準を満たした通信リンクを設定する。	共有メモリ接続を禁止するため。
-gd dba	DBA にデータベース起動パーミッションを設定する。	DBA 以外のユーザが自らのデータベースを起動して、DBA として接続し、UNLOAD または DROP DATABASE 文を実行する、またはエンジンかサーバを停止する可能性があるため。
-gk dba	データベース・エンジンまたはサーバの停止パーミッションを DBA に設定する。	DBA 以外のユーザが、データベース・エンジンまたはサーバを停止して、サービス拒否を引き起こす可能性があるため。
-gl dba	LOAD/UNLOAD パーミッションを DBA に設定する。	DBA 以外のユーザが、sybase ユーザのパーミッションを使用し、UNLOAD コマンドを使用してファイル・システムに書き込む可能性があるため。
-gu dba	ユーティリティ・コマンドのパーミッションを DBA に設定する。	DBA 以外のユーザが、DROP DATABASE 文を使用して、sybase ユーザが所有するデータベース・ファイルを削除する可能性があるため。

スイッチ	説明	理由
-x namedpipes(TDS=NO)	名前付きパイプ・ポートを起動し、TDS 接続を禁止する。	名前付きパイプ・ポートは、C2 基準を満たした構成でサポートされている唯一の通信メカニズムであり、TDS プロトコルは C2 基準を満たした構成に含まれていないため。

エンジンまたはサーバのオプション・スイッチ：

スイッチ	説明	制限
-a *logfile*	指定したトランザクション・ログ・ファイルを適用する。	リカバリのみで使用。
-b	バルク・オペレーション・モードで実行する。	
-c *size*	初期キャッシュの最大サイズを *size* バイトに設定する。	
-ca 0	メモリ割り付けを補う自動キャッシュ増加機能を無効にする。	
-ch *size*	キャッシュの最大サイズを *size* バイトに設定する。	
-cl *size*	キャッシュの最小サイズを *size* バイトに設定する。	
-cs	キャッシュ・サイズの統計を表示する。	
-ct	クライアントとエンジンまたはサーバ間の文字変換を実行する。	
-d	非同期 I/O を無効にする。	
-ec	通信メッセージを暗号化する。	
-f	トランザクション・ログなしでデータベースを強制的に起動する。	リカバリのみで使用。このスイッチを使用してエンジンまたはサーバを起動すると、監査が使用できないことに注意してください。

スイッチ	説明	制限
-ga	最後のデータベースが閉じられた後で自動的にシャットダウンする。	
-gc *num*	チェックポイント・タイムアウト時間を *num* 分に設定する。	
-ge *size*	外部 DLL スレッドのスタック・サイズを設定する。	
-gf	トリガの起動を無効にする。	
-gm *num*	可能であれば最大 *num* 個の接続を許可する。	
-gn *num*	*num* 個のエンジンまたはサーバ・スレッドを使用する。	
-gp *size*	最大ページ・サイズを *size* バイトに設定する。	
-gr *num*	最大リカバリ時間を *num* 分に設定する。	
-gt *num*	同時に実行する OS スレッドを *num* 個まで許可する。	
-gw *num*	バックグラウンド処理を *num* ミリ秒ごとに設定する。デフォルトは 500 ミリ秒。	
-gx *num*	*num* 個の OS スレッドを使用する。	
-m	チェックポイント後にトランザクション・ログをトランケートする。	これはチェックポイント後の監査ログもトランケートすることに注意してください。
-n *name*	データベース・エンジンまたはサーバの名前を指定する。	
-o *file*	メッセージ・ウィンドウのコピーを保存するファイル名。	

スイッチ	説明	制限
-os *size*	-o で指定されたファイルの最大サイズ。	
-p *size*	通信パケットの最大サイズを設定する。	
-q	クワイエット・モード - 出力を行わない。	
-r	読み込み専用モード - データベースの変更はできない。	
-ti *min*	接続が切断されるまでのクライアントのアイドル時間。デフォルトは 240 分。	
-tl *sec*	クライアントの活性タイムアウト (秒単位)。	C2 基準を満たした構成では効果はない。
-tq *time*	終了時刻を設定する。	
-u	バッファ・ディスク I/O を使用する。	
-v	製品バージョン情報を表示する。	
-z	デバッグ情報を表示する。	
-zo *file*	要求レベル・ログ情報をファイルにリダイレクトする。	
-zr *level*	要求レベル・ログを設定する。level は ALL、SQL、または NONE。	
-zs *size*	-zo に指定されたファイルの最大サイズ。	

db-file は、完全に修飾されたデータベース・ファイルまたはライト・ファイルの名前です。すべてのファイルは C2 データベースのフォルダに入れてください。

初期化ユーティリティ

構文　**dbinit -I** [*optional-switches*] *c2-database-folder¥filename*

必須スイッチ：

スイッチ	説明	理由
-I	jConnect サポートをインストールしない。	jConnect は TCP/IP を使用して通信するが、TCP/IP は C2 基準を満たした構成ではサポートされていないため。

オプション・スイッチ：

スイッチ	説明	制限
-b	比較のために文字列にブランクを埋め込む。	
-c	すべての文字列比較で大文字と小文字を区別する。	
-e	データベースを暗号化する。	
-m *name*	トランザクション・ログ・ミラー名を設定する。	フル・パスを指定し、ファイルは C2 データベースのフォルダに入れる。
-o *file*	ファイルに出力メッセージのログを取る。	
-p *size*	ページ・サイズを設定する。	
-q	クワイエット：メッセージを出力しない。	
-t *name*	トランザクション・ログ・ファイル名	フル・パスを指定し、ファイルは C2 データベースのフォルダに入れる。
-z *cs*	照合順を指定する。	

サービス作成ユーティリティ

構文 1 **dbsvc** [*optional-switches*] **-d** *svc name*

構文 2 **dbsvc** [*optional-switches*] **-a sybase** [*creation-switches*] **-w** *svc-name Details*

構文 3 **dbsvc** [**-q**] **-d** *svc name*

構文 4 **dbsvc** [**-q**] **-l**

必須スイッチ：

スイッチ	説明	理由
-a sybase	使用するアカウント名	Adaptive Server Anywhere サービスは sybase ユーザで実行するため。

オプション・スイッチ：

スイッチ	説明	制限
-q	バナーを表示しない。	
-y	確認メッセージを表示せずにサービスを削除または上書きする。	

作成スイッチ：

スイッチ	説明	制限
-p *passwd*	sybase アカウントのパスワードを指定する。	
-s *startup*	起動オプション。startup は Automatic、Manual、または Disabled。デフォルトは Manual。	
-t *type*	サービスのタイプ。type は Network または Standalone。デフォルトは Standalone。	

注意

構文 2 では、*Details* には、Adaptive Server Anywhere エンジンまたはサーバのフル・パスと、そのエンジンまたはサーバに対するパラメータを指定してください。

エンジンおよびサーバ・パラメータの詳細については、「データベース・エンジン／サーバ」103 ページを参照してください。

トランザクション・ログ・ユーティリティ

構文 **dblog** [*optional-switches*] *c2-database-folder¥database-file*

オプション・スイッチ：

スイッチ	説明	制限
-g *n*	LTM 世代番号を設定する。	
-il	LTM のトランケーション・ポイントを無視する。	
-ir	SQL Remote のトランケーション・ポイントを無視する。	
-m *name*	トランザクション・ログ・ミラー名を設定する。	フル・パスを指定し、ファイルは C2 データベースのフォルダに入れる。
-o *file*	ファイルに出力メッセージのログを取る。	
-q	クワイエット：メッセージを出力しない。	
-r	トランザクション・ログ・ミラーを使用しない。	
-t *name*	トランザクション・ログ名を設定する。	フル・パスを指定し、ファイルは C2 データベースのフォルダに入れる。
-x *n*	トランザクション・ログの現在の相対オフセットを *n* にする。	
-z *n*	トランザクション・ログの開始オフセットを *n* にする。	

Interactive SQL ユーティリティ

構文 1　**dbisqlc** [*optional-switches*] *SQL-command*

構文 2　**dbisqlc** [**optional switches**] *filename*

オプション・スイッチ：

スイッチ	説明	制限
-c *conn_str*	接続文字列 *conn_str* を使用する。	*conn_str* には "INT=YES;LINKS=namedpipes" を指定し、"UID=" または "PWD=" は指定しない。
-d *delimiter*	コマンド・デリミタを指定する。	
-q	サイレント・モード (ウィンドウの非表示)。	
-x	構文チェックのみ (コマンドは実行しない)。	

第 7 章

統合化ログイン

この章の内容

この章では、C2 基準を満たした構成と同じ方法による統合化ログインの使用について説明します。

統合化ログインの使用方法

Adaptive Server Anywhere では、統合化ログイン・メカニズムを使用して Windows ユーザを Adaptive Server Anywhere ユーザにマッピングしています。Windows ユーザがデータベースに接続しようとすると、そのユーザが認証されていることが (通常はパスワードを使用して) オペレーティング・システムによって保証されます。その Windows ユーザと有効な Adaptive Server Anywhere ユーザのマッピングがデータベース・サーバに存在する場合、そのユーザは接続できます。

C2 基準を満たした構成で使用するためには、Adaptive Server Anywhere では統合化ログインのみを使用する必要があります。統合化ログインを、データベースの DBA アカウントに対して作成してください。Windows ユーザ sybase をこの目的で使用することをおすすめします。さらに、統合化ログイン・マッピングは 1 対 1 にします。つまり、2 つの Windows ユーザ・アカウントを同じ Adaptive Server Anywhere アカウントにマッピングすることはできません。

sybase ユーザの統合化ログインを作成する方法の手順については、「データベースの作成」67 ページを参照してください。

統合化ログインの詳細については、『ASA データベース管理ガイド』>「データベースへの接続」を参照してください。

第 8 章

Adaptive Server Anywhere サービスへの接続

この章の内容

この章では、C2 基準を満たした構成と同じ方法による Adaptive Server Anywhere サービスへの接続について説明します。

Adaptive Server Anywhere サービスへの接続

Adaptive Server Anywhere サービスがいったん開始すると、ユーザは dbisqlc を使用してエンジンに接続し、SQL 文を実行できます。dbisqlc で接続方法を指定するには次の 2 つの方法があります。

1. -c スイッチを使用すると接続文字列を指定できます。接続文字列は、接続先のエンジンとデータベース、およびその検索方法を dbisqlc に指定するパラメータのリストです。たとえば、エンジンの名前が asademo の場合は次の文字列を使用して接続できます。

   ```
   dbisqlc -c "ENG=asademo;LINKS=namedpipes;INT=YES
   ```

   LINKS=namedpipes によって、名前付きパイプを使用してエンジンに接続することを dbisqlc に指定します。また、INT=YES によって、統合化ログイン機能を使用することを dbisqlc に指定します。

2. スイッチを指定しないで単に dbisqlc を開始して、[接続] ダイアログでフィールドを設定することもできます。[ログイン] タブの [統合化ログインの使用] オプションを選択し、[データベース] タブにサーバ名を入力して、[ネットワーク] タブの [名前付きパイプ] チェックボックスをオンにしてください。

第 9 章

Adaptive Server Anywhere の C2 パッチ

この章の内容

この章では、Adaptive Server Anywhere リリース 7.0.0 の C2 パッチについて説明します。この章の内容は、現在のソフトウェアを C2 基準を満たした環境と同じ方法で実行している場合には適用されません。

Adaptive Server Anywhere の C2 パッチ

Adaptive Server Anywhere の C2 パッチには、2 つの DLL、3 つのバッチ・ファイル、1 つのテキスト・ファイルが含まれています。この項では、パッチの各ファイルについて説明します。

この項では、Adaptive Server Anywhere リリース 7.0.0 の C2 パッチについて説明します。この項の内容は、現在のソフトウェアを C2 基準を満たした環境と同じ方法で実行している場合には適用されません。

ファイル	説明
dblgen7.dll	Adaptive Server Anywhere エンジンとツールで使用する英語の文字列が含まれます。このファイルには、-xo スイッチの使用を監査する dbbackup で使用される監査文字列が含まれます。
dbtool7.dll	すべてのデータベース・ユーティリティと *dbisqlc.exe* によって使用されます。このファイルには、トランザクション・ログのトランケーションを監査する dbbackup の修正が含まれます。
mdac1.bat	このバッチ・ファイルを実行するとテンポラリ・ディレクトリが作成され、そこに 14 個の *.dll* ファイルと 1 つの *.exe* が *C:¥winnt¥system32* ディレクトリからコピーされます。Adaptive Server Anywhere をインストールすると、これらのファイルが置換されます。ファイルを後でリストアできるように、ファイルをコピーしてから Adaptive Server Anywhere をインストールしてください。
mdac2.bat	このバッチ・ファイルを実行すると、*mdac1.bat* で作成されたテンポラリ・ディレクトリにあるファイルが *C:¥winnt¥system32* ディレクトリにコピーされ、Adaptive Server Anywhere がインストールしたファイルを上書きします。ファイルの 1 つはオペレーティング・システムによって使用されているため、名前が変更されてからコピーされます。
mdac3.bat	このバッチ・ファイルを実行すると、*mdac2.bat* で名前を変更したファイルと、*mdac1.bat* で作成したテンポラリ・ディレクトリが削除されます。

ファイル	説明
readme.txt	このファイルには、パッチの *.dll* ファイルをインストールするための指示があります。

第 10 章

その他の情報

この章の内容

この章では、C2 基準を満たした構成で Adaptive Server Anywhere を運用するときに役立つ追加情報リストを示します。

その他の情報の参照先

このマニュアルの内容	ソース
監査	「安全なデータの管理」3 ページ
接続パラメータ	リストについては、dbdsn -cl を実行するか、『ASA データベース管理ガイド』>「クライアント／サーバ通信」を参照
データベース・オプション	『ASA データベース管理ガイド』>「データベース・オプション」
dbinit、dblog、dbtran、dbisqlc、dbbackup、その他の管理ユーティリティ	『ASA データベース管理ガイド』>「データベース管理ユーティリティ」
dbsvc ユーティリティ	『ASA データベース管理ガイド』>「サービス作成ユーティリティ」
エンジン／サーバのスイッチ	『ASA データベース管理ガイド』>「データベース・サーバ」
統合化ログイン	『ASA データベース管理ガイド』>「データベースへの接続」
データベースの Java	『ASA プログラミング・ガイド』>「データベースにおける Java の概要」および『ASA プログラミング・ガイド』>「データベースにおける Java の使用」
プロシージャ、関数、トリガ	『ASA SQL ユーザーズ・ガイド』>「プロシージャ、トリガ、バッチの使用」
セキュリティのヒント	「安全なデータの管理」3 ページ
テーブル、ビュー	『ASA SQL ユーザーズ・ガイド』>「データベース・オブジェクトの使用」
GRANT と REVOKE の SQL 文	『ASA SQL リファレンス・マニュアル』>「SQL 文」
トランザクション・ログ・ファイル	『ASA データベース管理ガイド』>「バックアップとデータ・リカバリ」

このマニュアルの内容	ソース
ユーザ ID とパーミッション	『ASA データベース管理ガイド』>「ユーザ ID とパーミッションの管理」

索引

あ

え

お

か

き

す

せ

て

と

に

ね

は

ひ

ふ

ま

み

ゆ

よ

ら

る

れ

ろ